Pragmatic Security Automation in the Cloud with **Python**

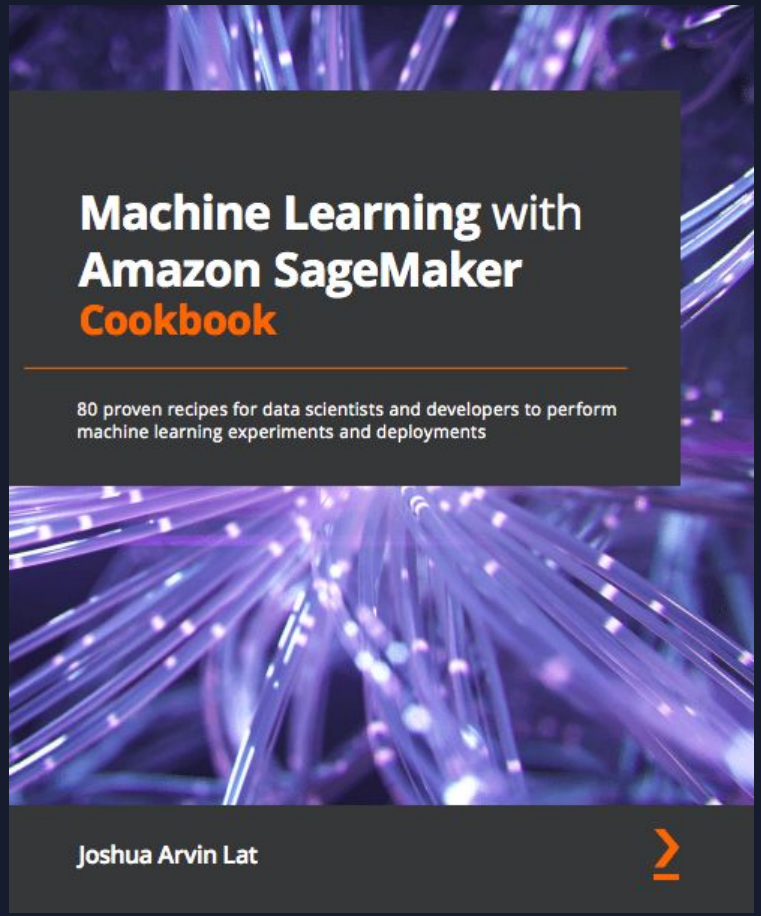
•••

aws

machine learning

HERO

- **Chief Technology Officer** of NuWorks Interactive Labs
- AWS Machine Learning Hero
- Author of 📖 **Machine Learning with Amazon SageMaker Cookbook**



Author of 📖

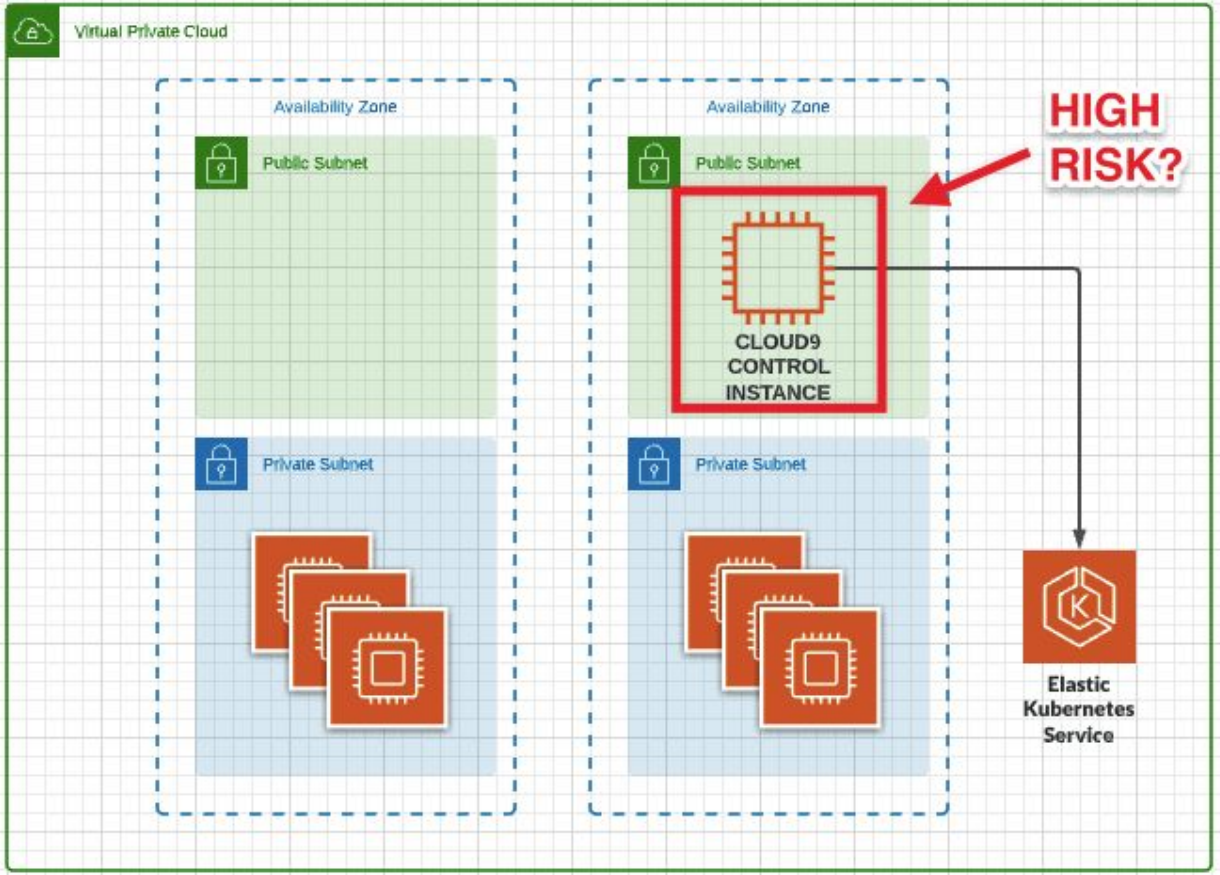
Machine Learning with Amazon SageMaker Cookbook

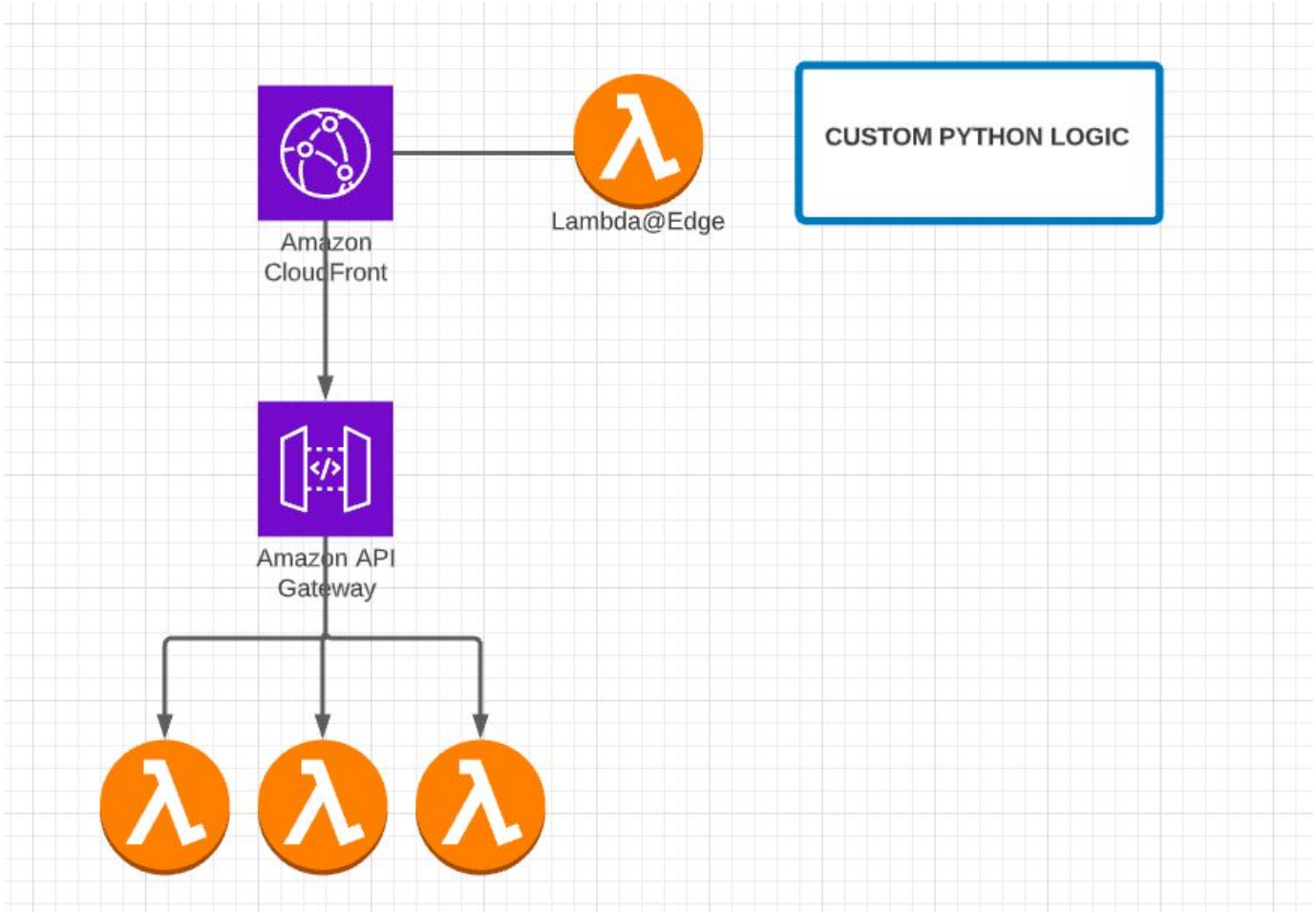
80 proven recipes for data scientists and developers to perform machine learning experiments and deployments

Pragmatic Security Automation in the Cloud with Python

- CYBERSECURITY ATTACK CHAIN
- SECURITY AUTOMATION WITH A PURPOSE
- SECURITY AUTOMATION SCRIPTING TIPS
- AUTOMATED DATA INTEGRITY LAYER
- AUTOMATED VULNERABILITY MANAGEMENT
- (SECURE) INFRASTRUCTURE AS CODE

CYBERSECURITY ATTACK CHAIN


Virtual Private Cloud
Availability Zone
Public Subnet
Private Subnet
Availability Zone
Public Subnet
Private Subnet
HIGH RISK?
CLOUD9 CONTROL INSTANCE
Elastic Kubernetes Service


Amazon CloudFront
Lambda@Edge
CUSTOM PYTHON LOGIC
Amazon API Gateway

SECURITY AUTOMATION WITH A PURPOSE

+

+

<INSERT TOOL HERE>

CUSTOM PASSWORD CRACKING SCRIPT

PASSWORD PROTECTED ZIP FILE

SECURITY AUTOMATION SCRIPTING TIPS

sample.py

Raw

```python
import json
import uuid

from time import sleep
from contextlib import contextmanager


@contextmanager
def block(label):
    print(f"[{label}]: START")
    yield
    print(f"[{label}]: END")


def main():
    with block('LOAD PARAMETERS'):
        ...
```

```
class Node(object):
    def __init__(self, data):
        self.__internal_data = data

        if isinstance(data, dict):
            self.__process_dictionary(data)

        if isinstance(data, list):
            self.__process_list(data)

    def __process_value(self, value):
        processed_value = None

        if isinstance(value, dict) or isinstance(value, list):
            processed_value = Node(value)
        else:
            processed_value = value

        return processed_value

    def __process_list(self, target):
        self.__list_contents = target
        self.__processed_list_contents = []

        for list_content in self.__list_contents:
            self.__processed_list_contents.append(self.__process_value(list_content))

        def item(internal_self, index):
            return internal_self.__processed_list_contents[index]

        self.item = MethodType(item, self)
```

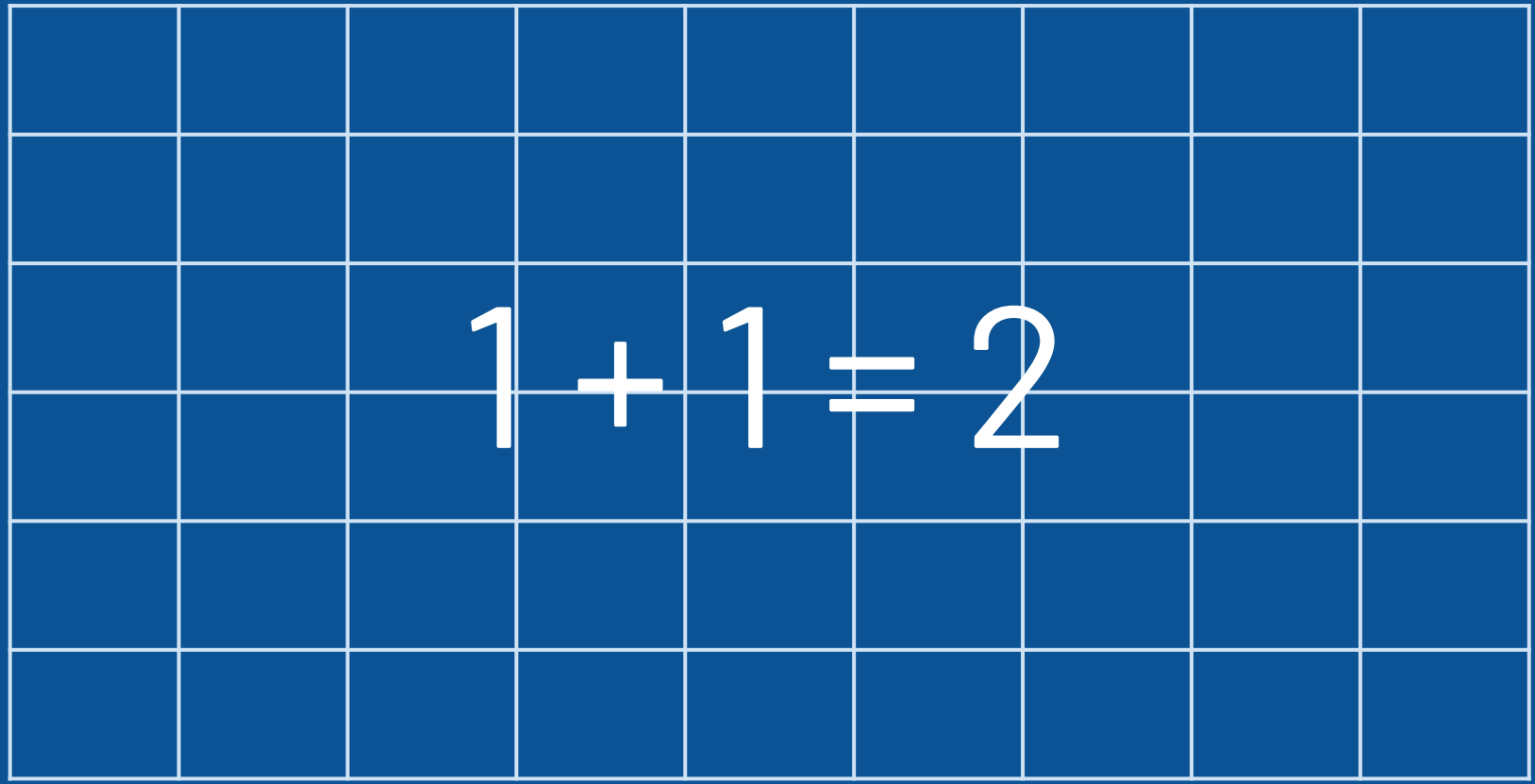
```
def __process_dictionary(self, target):
    self.__internal_keys = target.keys()

    for key, value in target.items():
        head, *tail = key.split('.')

        key = head

        if not tail:
            processed_value = self.__process_value(value)

        else:
            child_head_string = '.'.join(tail)

            processed_value = Node({ child_head_string: value })

        setattr(self, key, processed_value)

def __str__(self):
    return str(self.__internal_data)

def __repr__(self):
    return self.__str__()
```

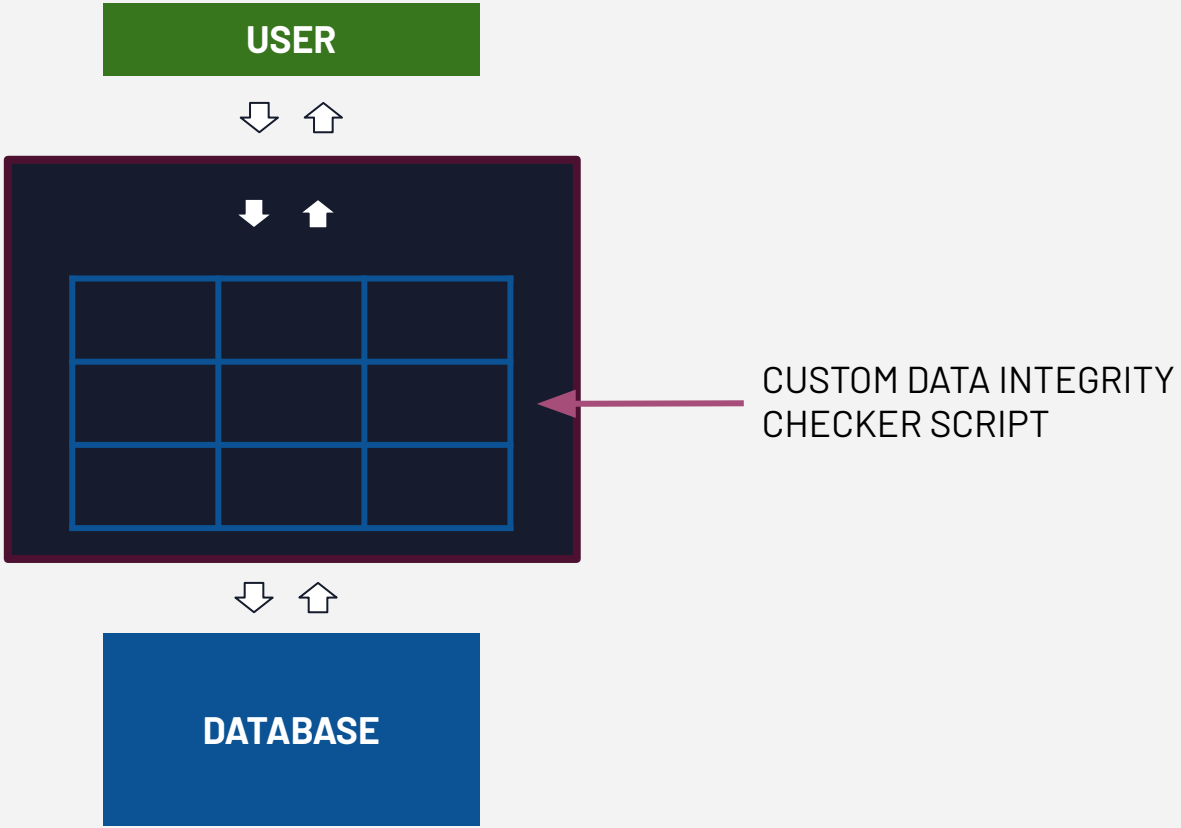
node.py

Raw

```python
from var_utils import Node

dictionary = {
    'mail': {
        'username': 'username@gmail.com',
        'password': 'password'
    },
    's3_file_uploader': {
        'local_download_folder': '/'
    }
}

node = Node(dictionary)

print(node.mail.username)
print(node.internal_keys)
print(node.internal_dictionary)
```

AUTOMATED DATA INTEGRITY LAYER

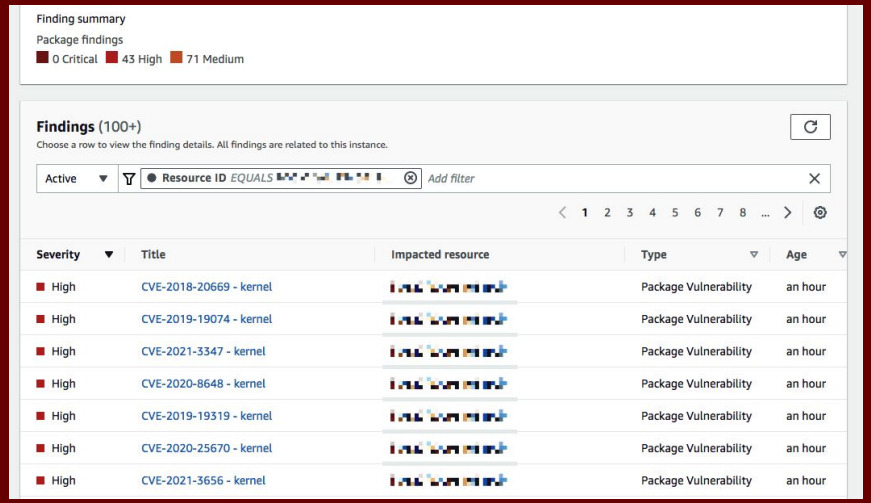

1+1=2



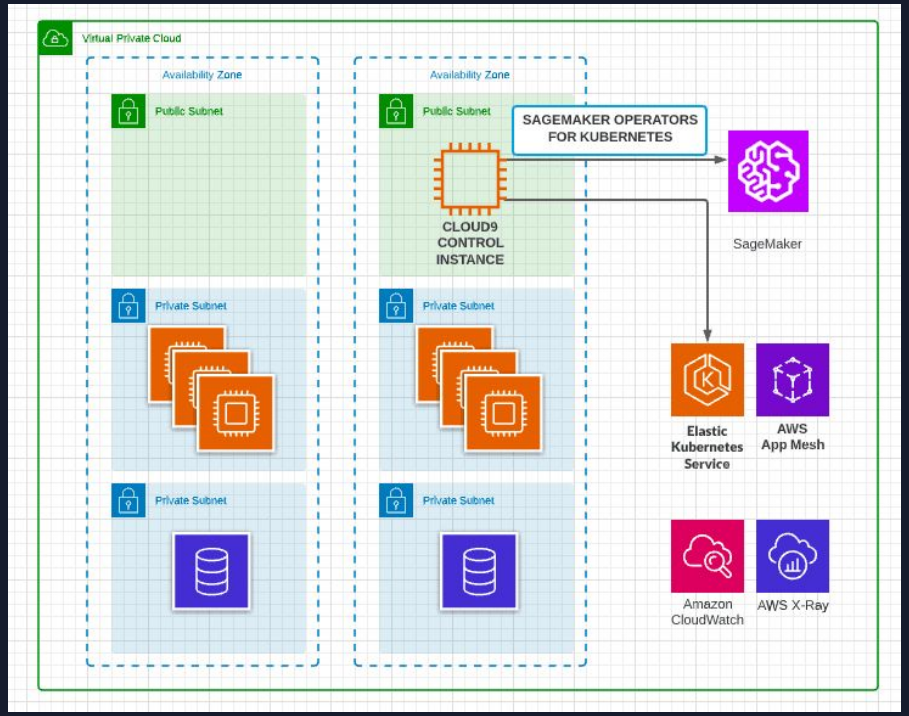
docker Flask SQLAlchemy

AUTOMATED VULNERABILITY MANAGEMENT

VULNERABILITY ASSESSMENT TOOL

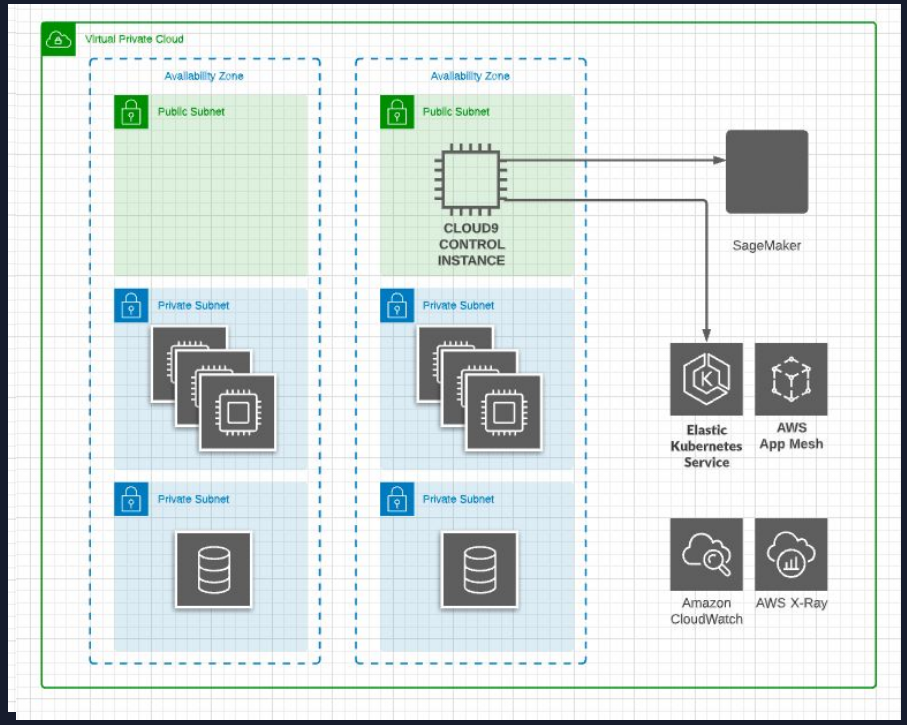


(SECURE)
INFRASTRUCTURE AS CODE



```
{
   'key': 'value'
}
```

boto3 + Python + CloudFormation

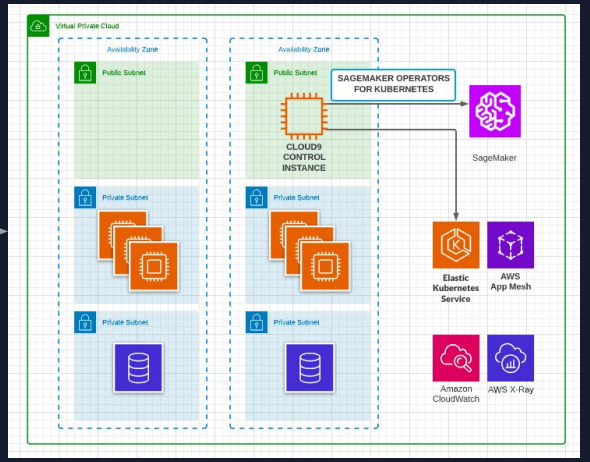


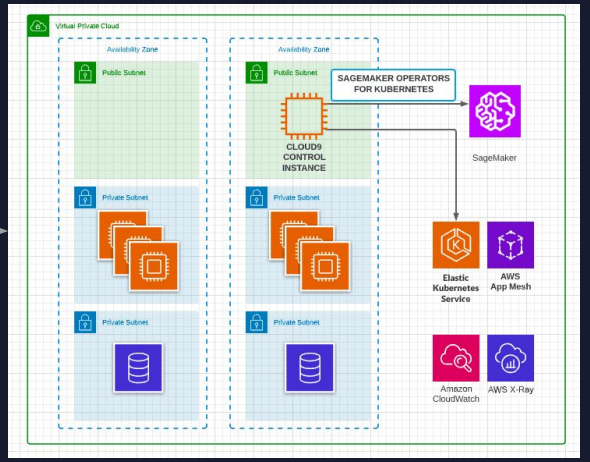

```
{
   'key': 'value'
}
```

boto3 + Python + CloudFormation

```
{
   'key': 'value'
}
```

boto3 + Python + CloudFormation





Pragmatic Security Automation in the Cloud with Python

- CYBERSECURITY ATTACK CHAIN
- SECURITY AUTOMATION WITH A PURPOSE
- SECURITY AUTOMATION SCRIPTING TIPS
- AUTOMATED DATA INTEGRITY LAYER
- AUTOMATED VULNERABILITY MANAGEMENT
- (SECURE) INFRASTRUCTURE AS CODE