

Real Time Trading

 LIVE

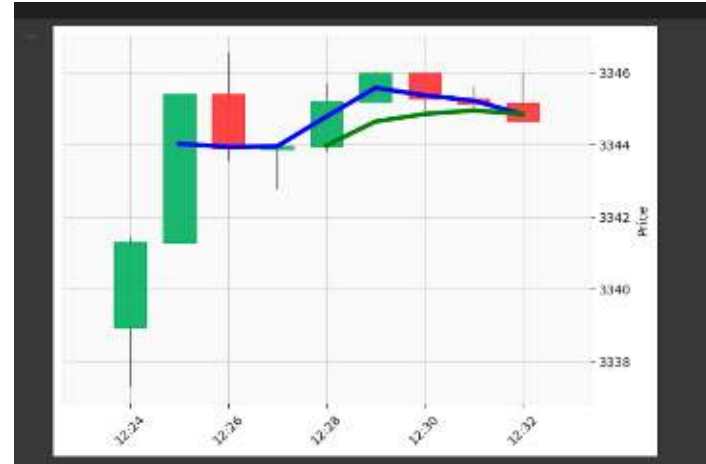
# PYTHON TRADING TRACK

---

พิเศษเพิ่มเนื้อหา ปล่อยระบบเทรดบน cloud

18 July เวลา 7:00 PM

# Algo Execution

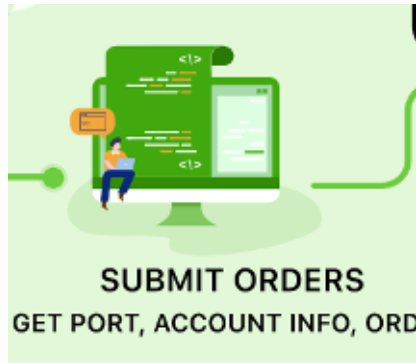


Date	Pair	Type	Side	Average	Price	Executed	Amount	Total
+ 03-28 19:33:00	ETH/BUSD	Market	Sell	3,344.71	Market	0.0060	0.0060	20.07





# Algorithmic Trading need both



**Executions**



**Strategies**



# Real Time Trading Covering

## Crypto

**Bitkub API**

**Binance API**

---

**CCXT**

---

***CCXT PRO***

## Thai Equity

**SET OPEN API**

→ **Stock**

→ **TFEX**

## Forex

**Interactive Broker**

→ **Forex**

→ **Foreign Stocks**

→ **Foreign Futures &  
Options / Bitcoin  
Futures**

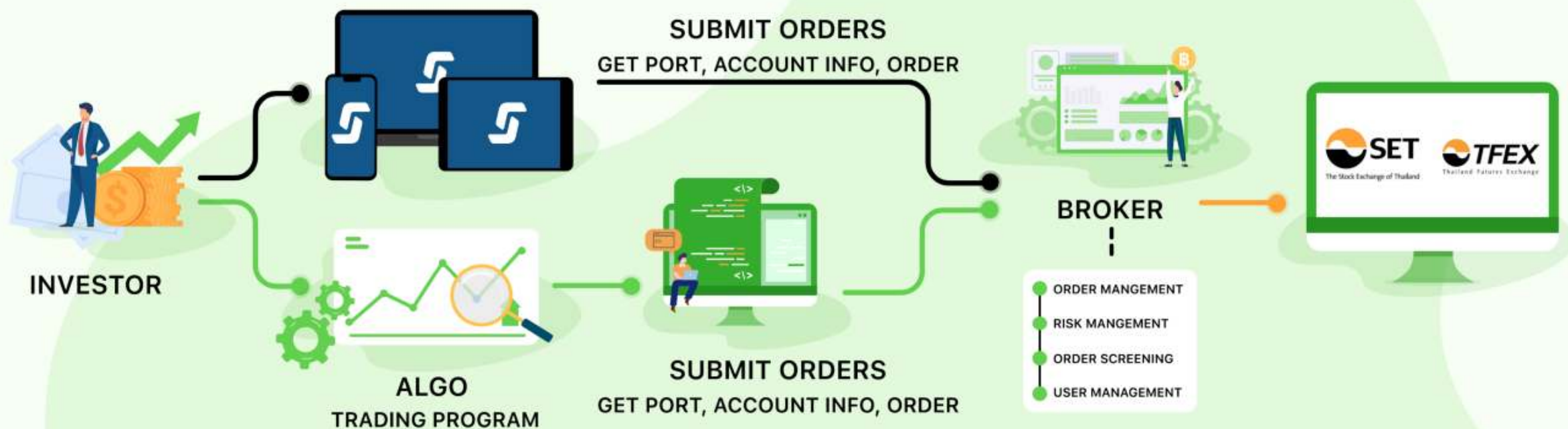
***Tradingview Python API  
Metatrader Wrapper***

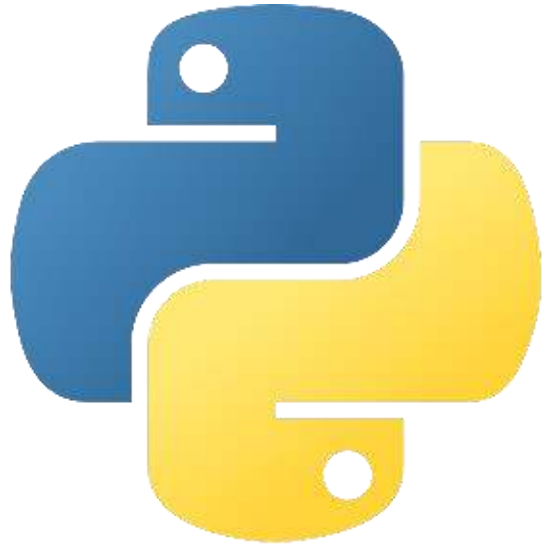


## Real Time Trading Agenda

- API Connection
- Account API
- Ordering API
- Building Block / Concept
- Execution Loop
- *Promotion from Brokers !*

# Application vs API



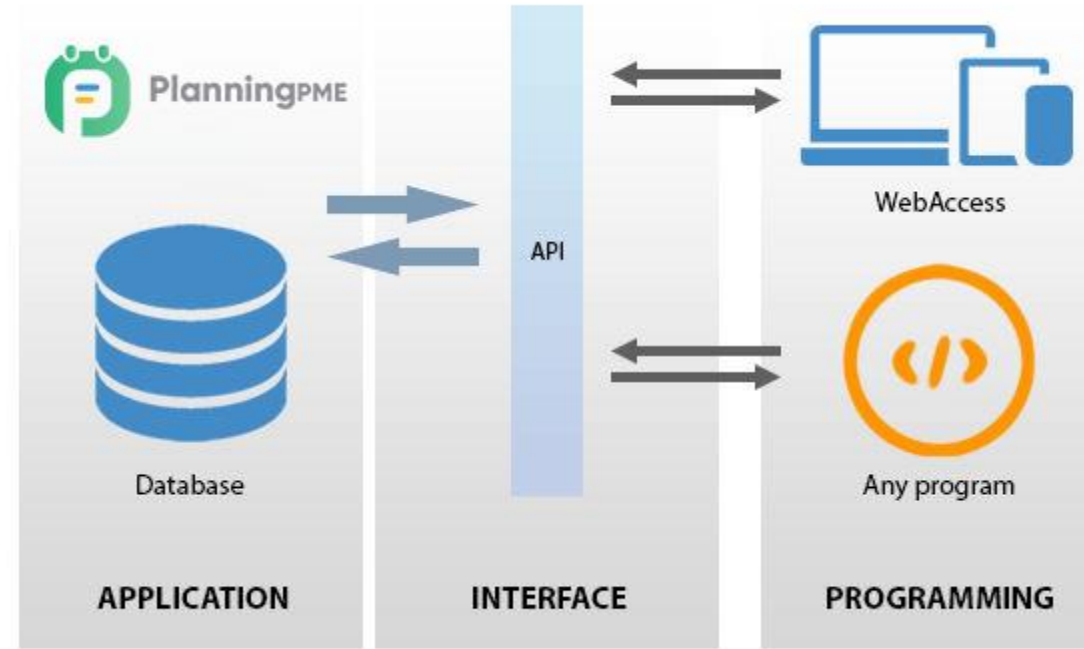


# API Connection

with Python

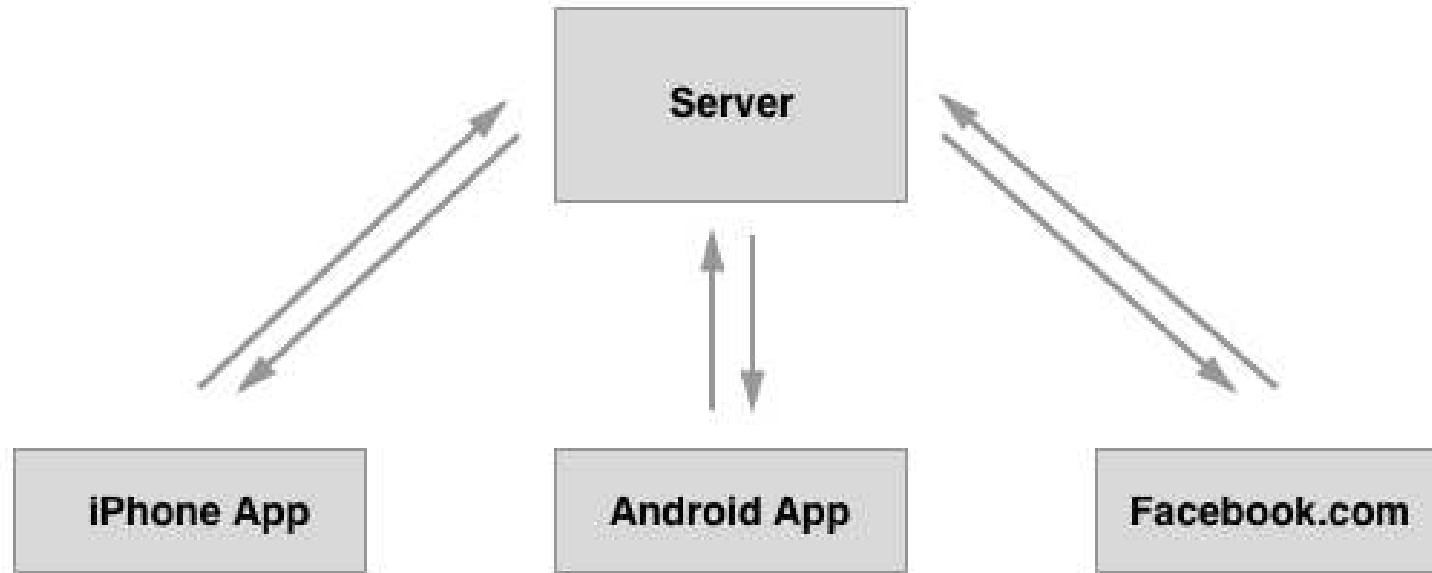


Imagine what we did before ... get data from yfinance

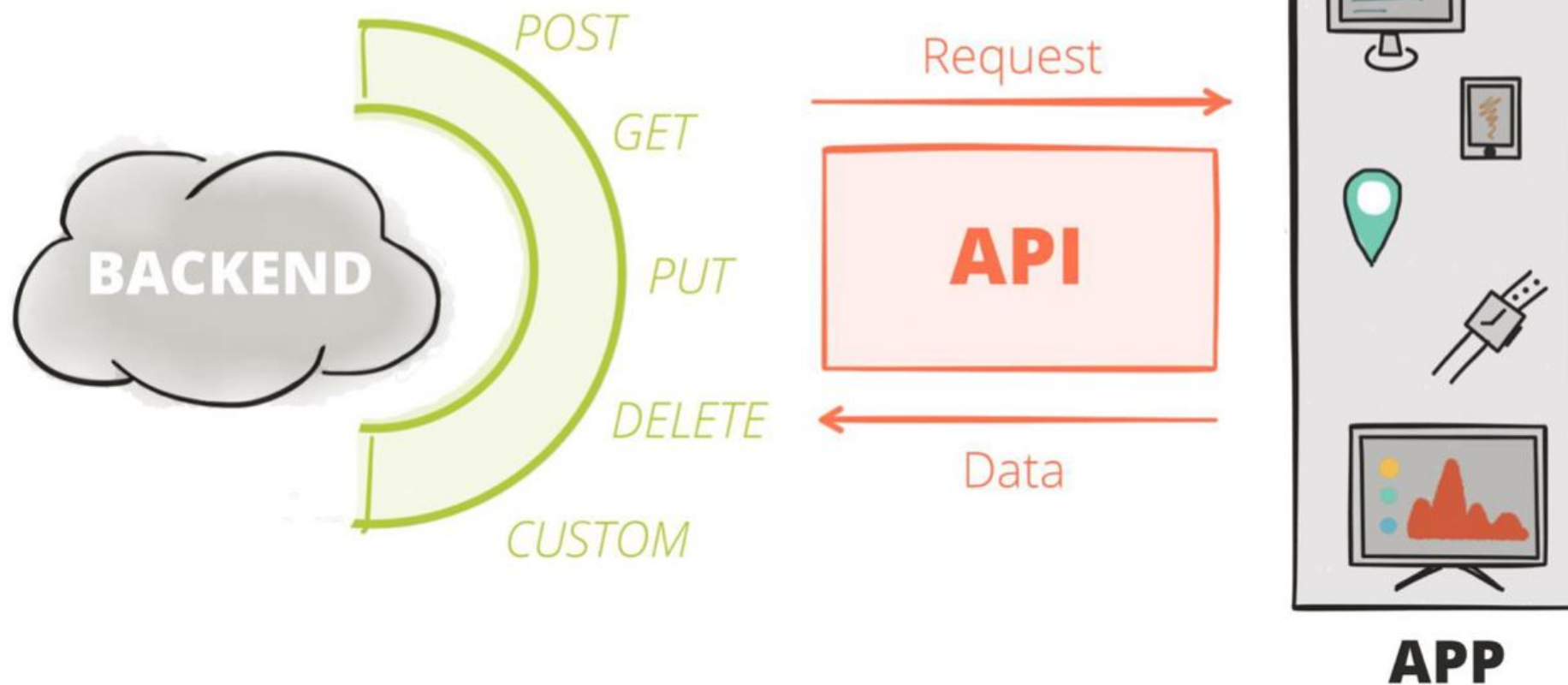


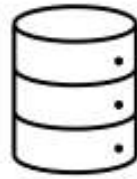


# What is API ?



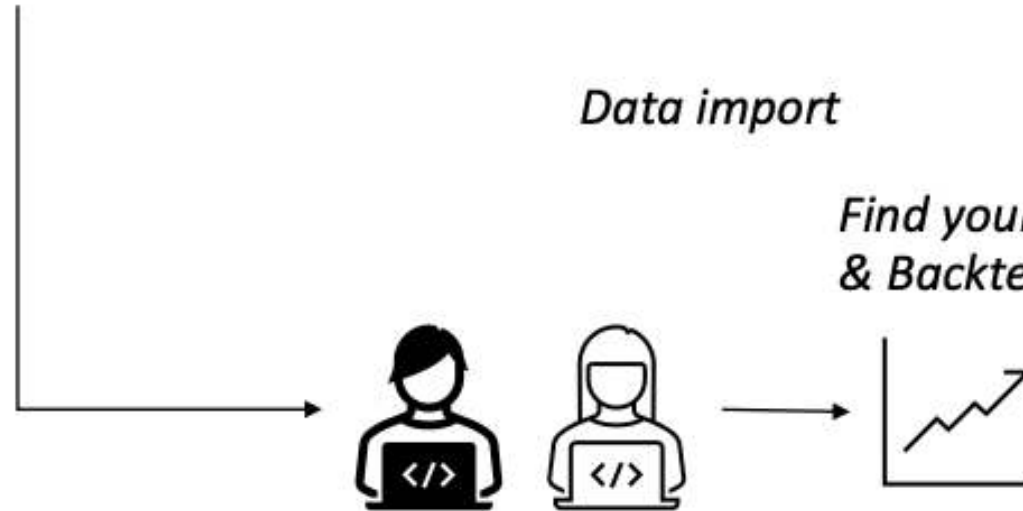
# What is API ?

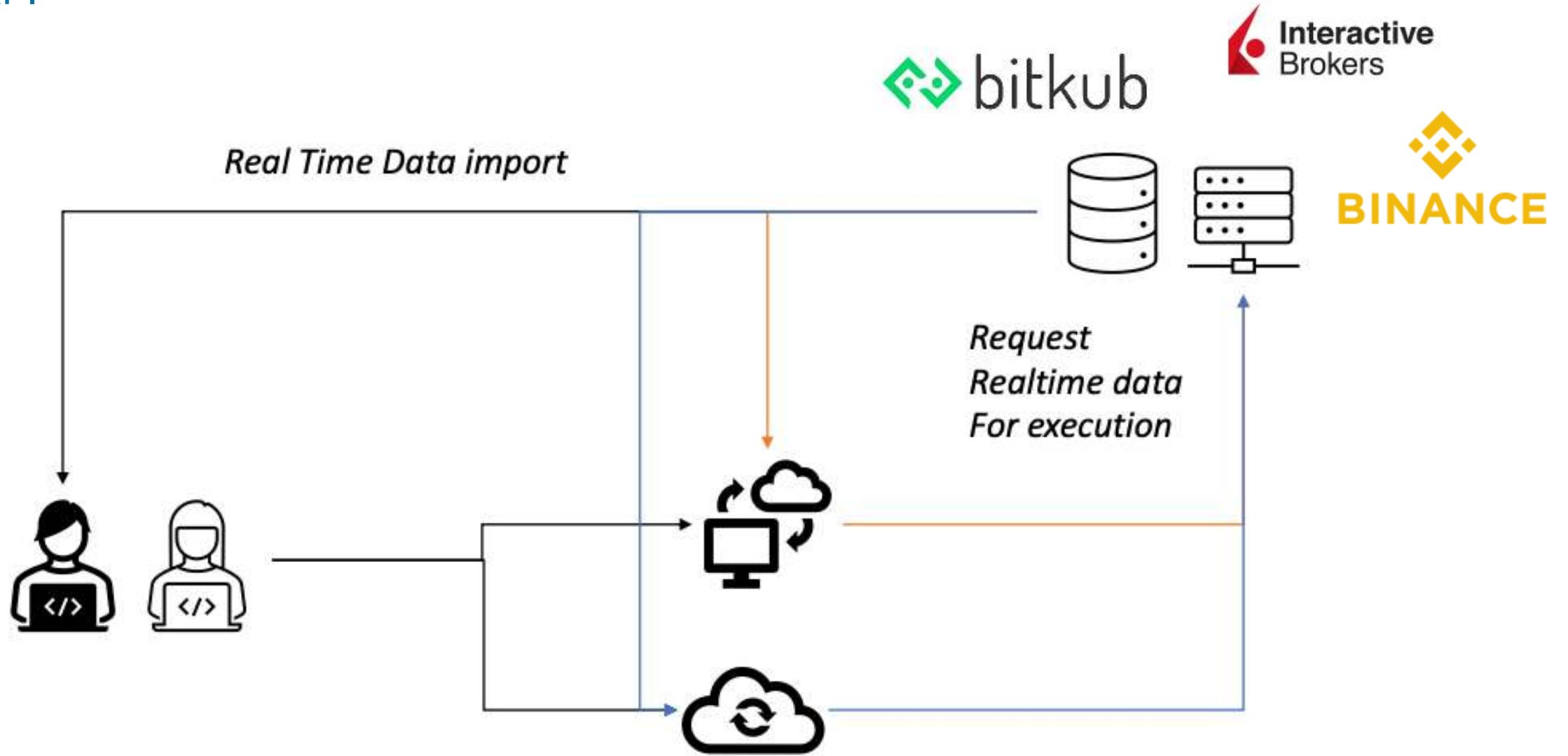




*Data import*

*Find your 'best strategy'  
& Backtesting*









```
!pip install python-binance
```

```
from binance.client import Client
```

```
api_key = 'o2[REDACTED]DLZ4CT4o  
api_secret = '[REDACTED]g3c3pfy6
```

By pass Login



```
order = client.order_limit_buy(  
                                symbol='BNBBUSD',  
                                quantity=0.10,  
                                price='380')
```

order

```
{'clientOrderId': 'QqmYrXmCwYtFMMUeqAOhKK',  
 'cumulativeQuoteQty': '0.00000000',  
 'executedQty': '0.00000000',  
 'fills': [],  
 'orderId': 1175558212,  
 'orderListId': -1,  
 'origQty': '0.10000000',  
 'price': '380.00000000',  
 'side': 'BUY',  
 'status': 'NEW',  
 'symbol': 'BNBBUSD',  
 'timeInForce': 'GTC',  
 'transactTime': 1648387930137,  
 'type': 'LIMIT'}
```

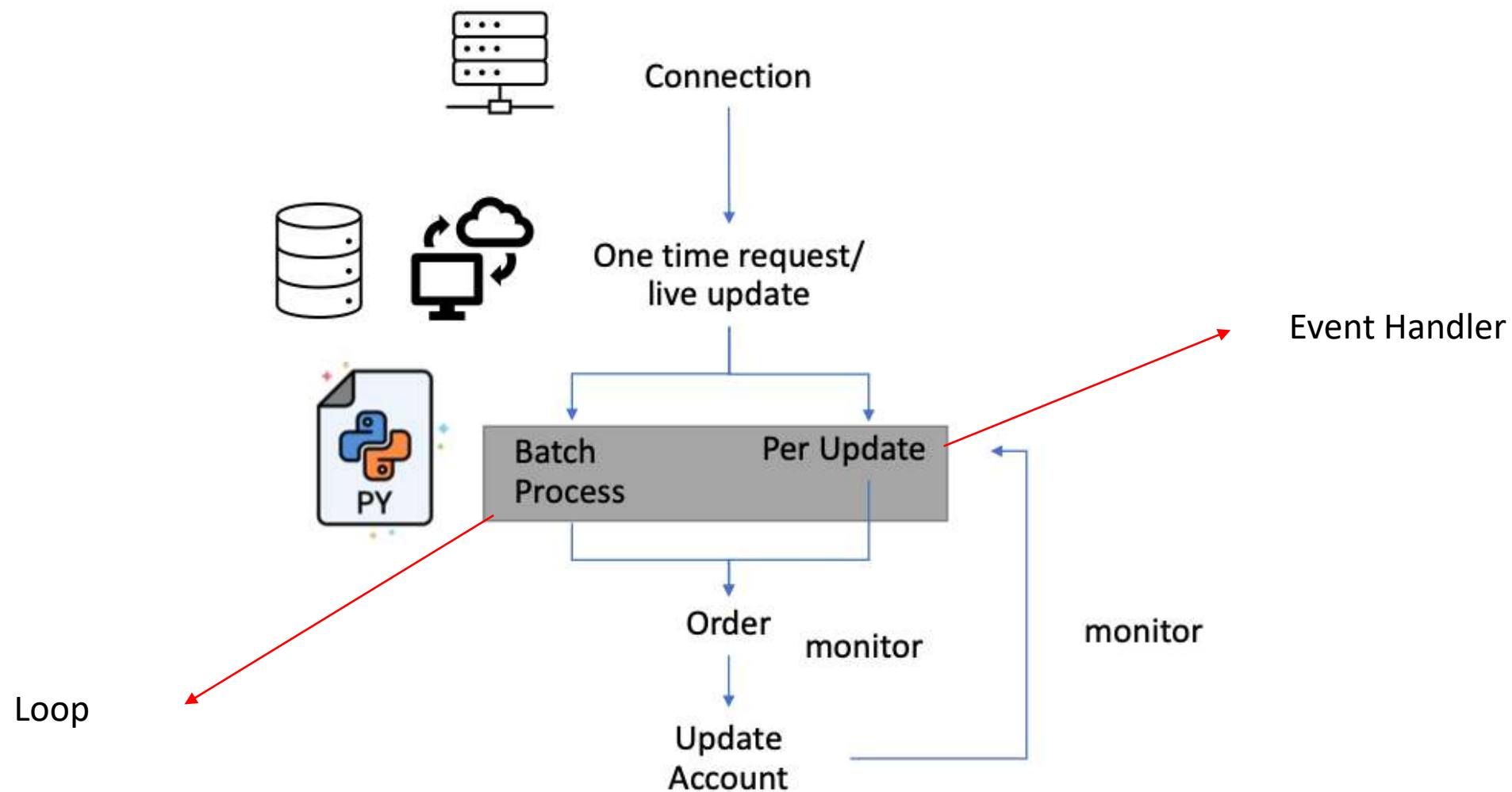


## In reality → super high level concept

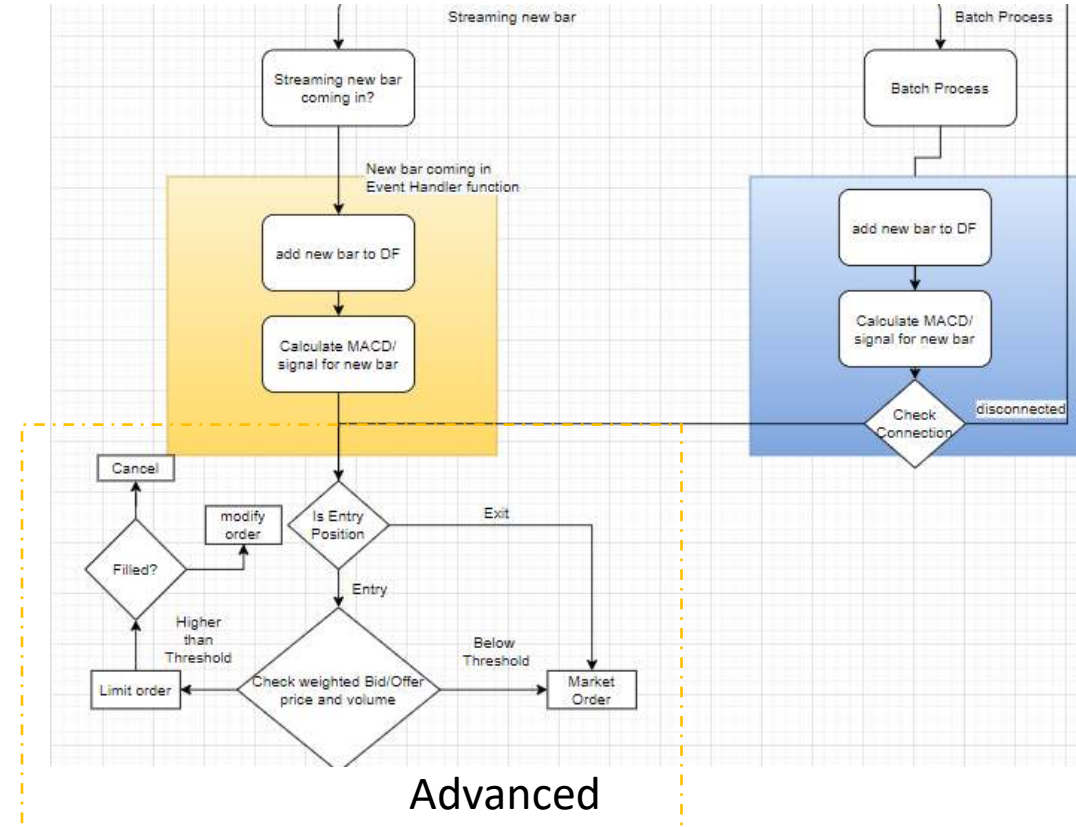
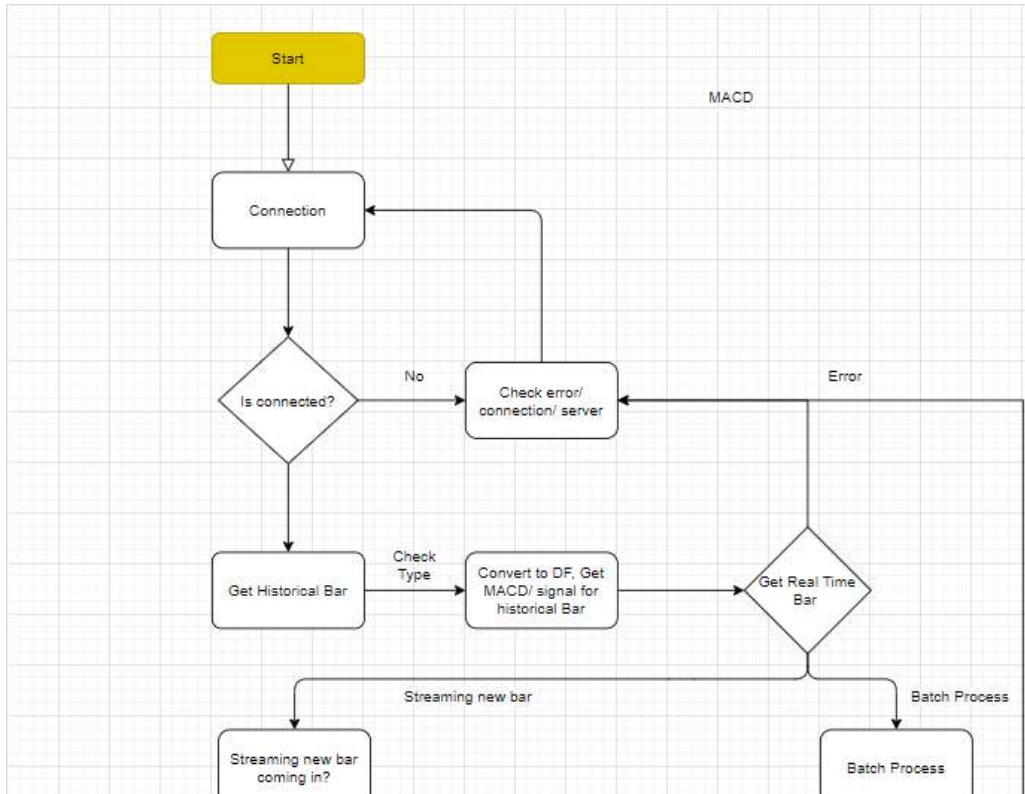
0	Import & Initialization
1	Authen → API Key & Secret
2	API Data Retrieve Somewhat cleaning, prepare data Signal Creation
3	API Order
4	Monitoring, Risk Management



# Building Block

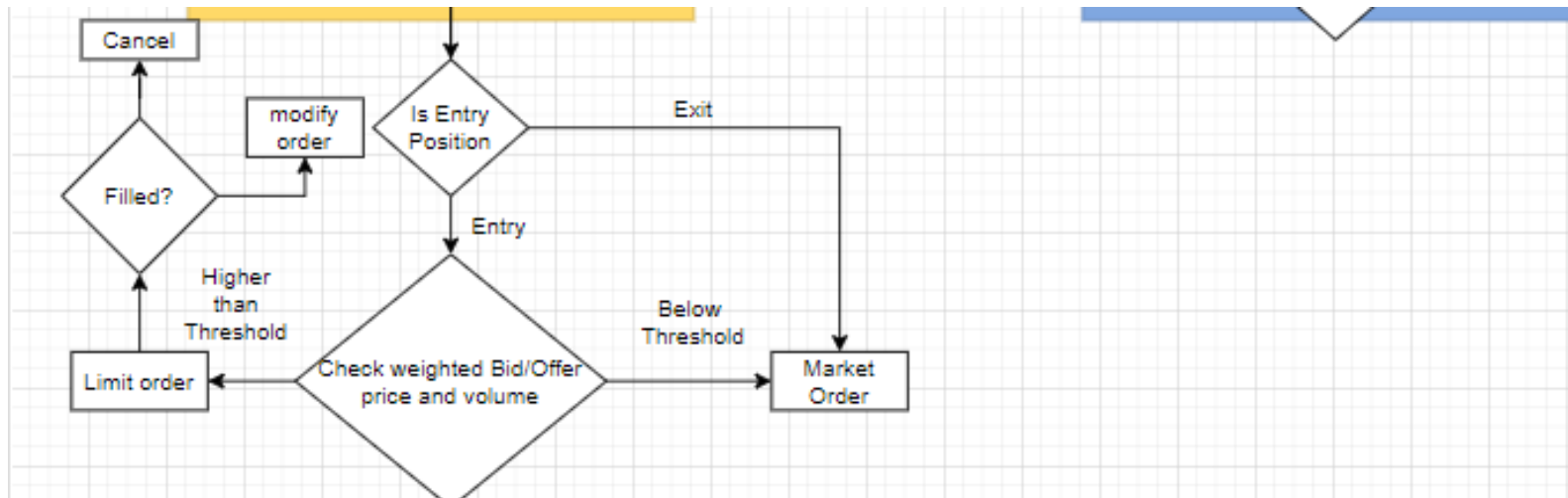


e.g. MACD with limit order for entry, MP order for exit












# Building Block

e.g. MACD with limit order for entry, MP order for exit



## รายชื่อ โบรกเกอร์ผู้ให้บริการ

ชื่อ	Derivatives API	Equity API	Derivatives Market Data API	Equity Market Data API
 Classic Ausiris (063)	✓	N/A	✓	N/A
 Finansia Syrus (024)	N/A	✓	N/A	✓
 Pi (003)	✓	N/A	✓	N/A
 KGI (013)	✓	✓	✓	✓
 Globlex (025)	✓	✓	✓	✓
 Kingsford (015)	N/A	✓	N/A	✓
 Krungsri (029)	✓	N/A	✓	N/A
 Yuanta (019)	✓	N/A	✓	N/A
 YLG (062)	✓	N/A	✓	N/A



As the friend of a referrer, you can earn up to \$1000 of IBKR stock, subject to program conditions.

 United States<sup>6</sup>

FIXED

No Transaction Fee ETFs

USD 0.005 per share

Example +

MINIMUM PER ORDER

USD 0.00

USD 1.00

MONTHLY TRADE AMOUNT <sup>2</sup>	COMMISSIONS	MINIMUM PER ORDER <sup>2</sup>
USD <= 1,000,000,000	0.20 basis point <sup>3</sup> * Trade Value <sup>4</sup>	USD 2.00
USD 1,000,000,001 - 2,000,000,000	0.15basis point <sup>3</sup> * Trade Value <sup>4</sup>	USD 1.50
USD 2,000,000,001 - 5,000,000,000	0.10basis point <sup>3</sup> * Trade Value <sup>4</sup>	USD 1.25
USD > 5,000,000,000	0.08basis point <sup>3</sup> * Trade Value <sup>4</sup>	USD 1.00



To be Continue ....



# What's New?

# What is Cloud ?

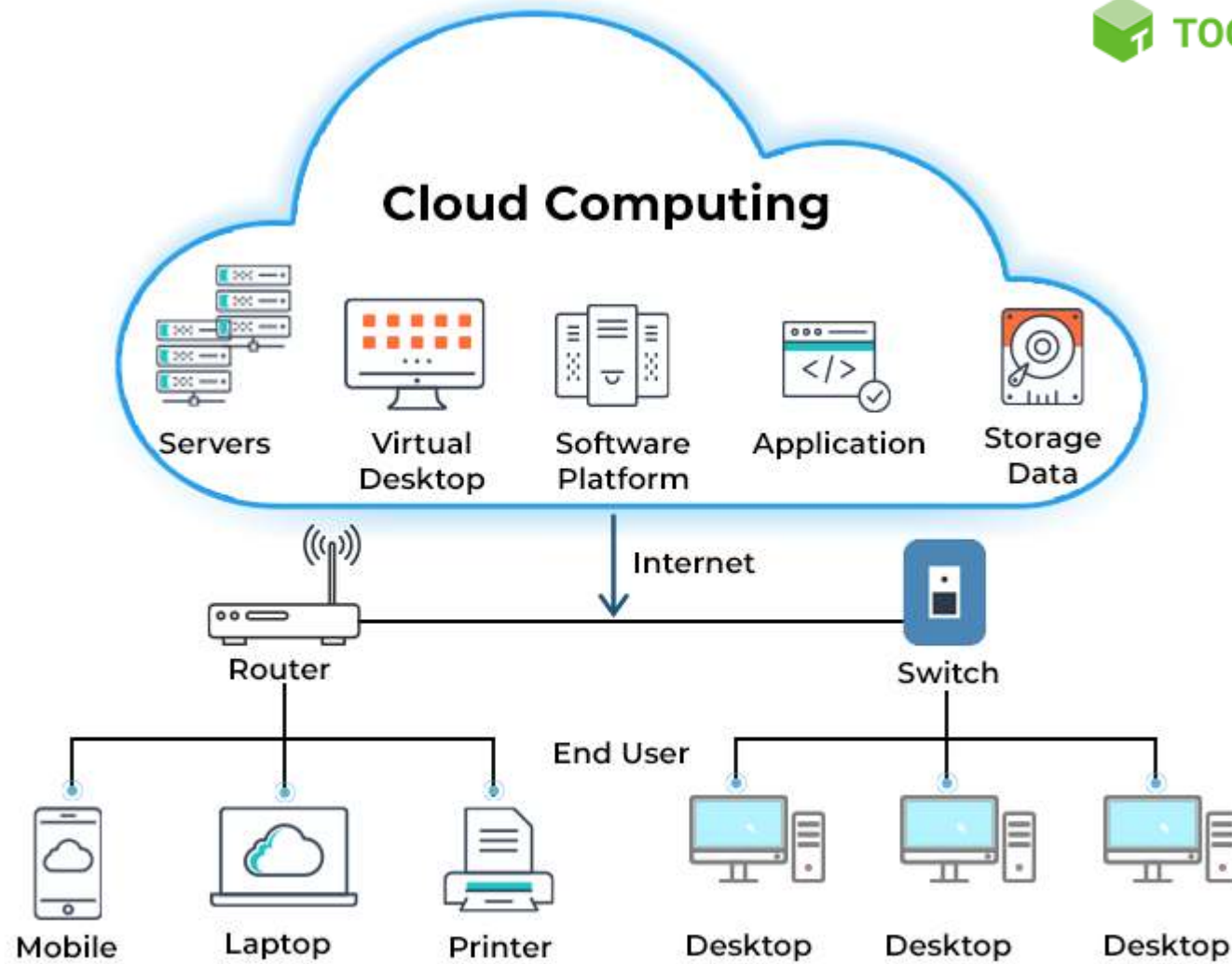


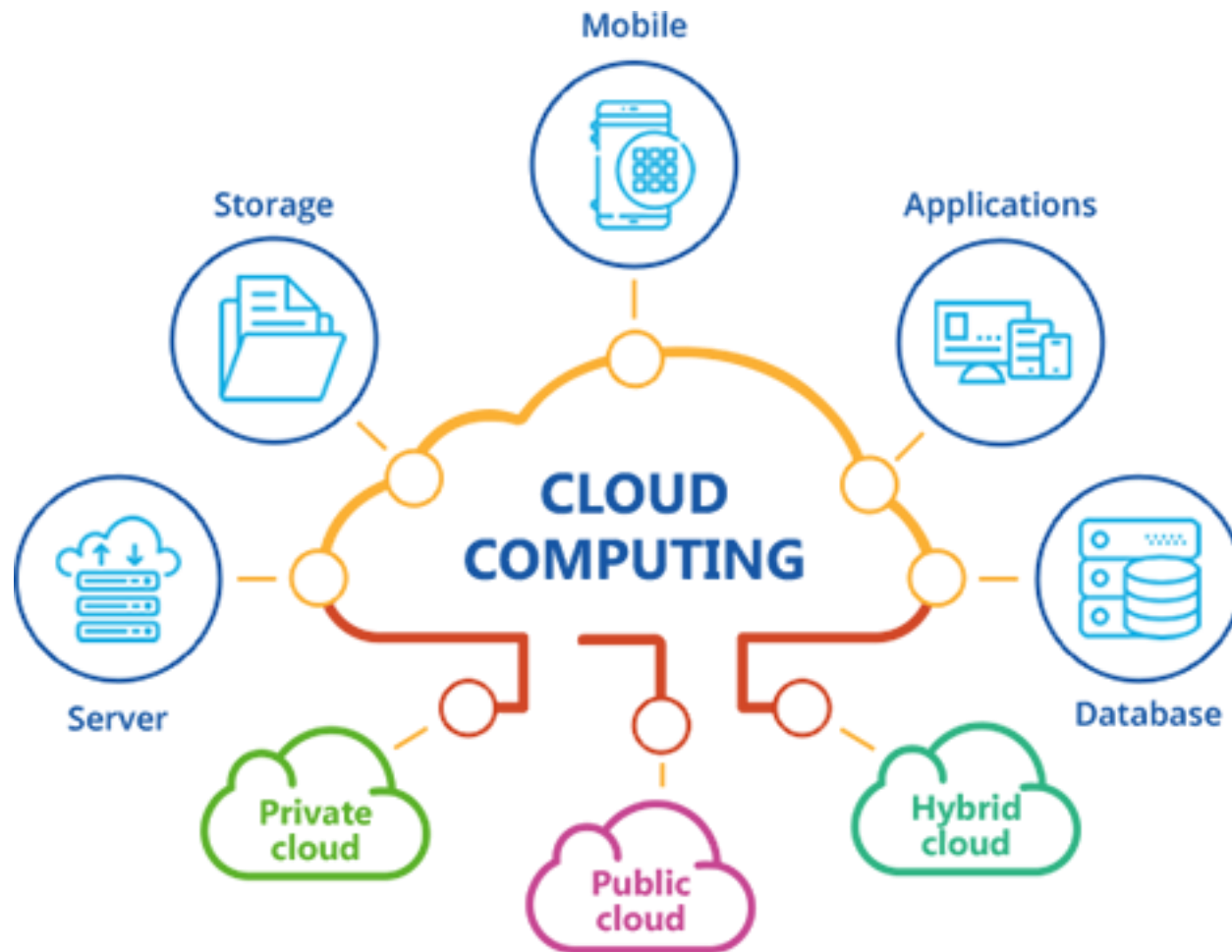
## What is Cloud ?





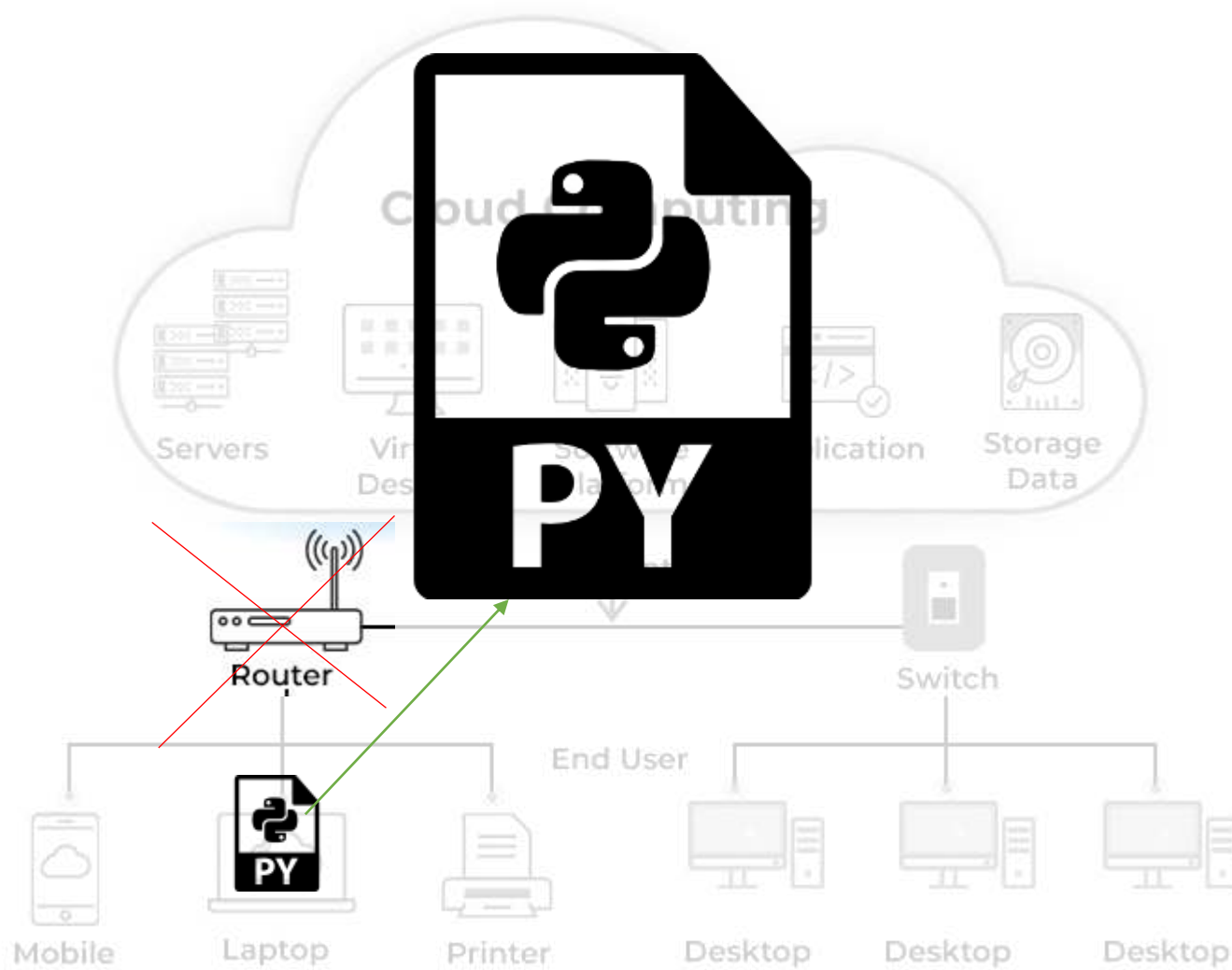
# What is Cloud ?







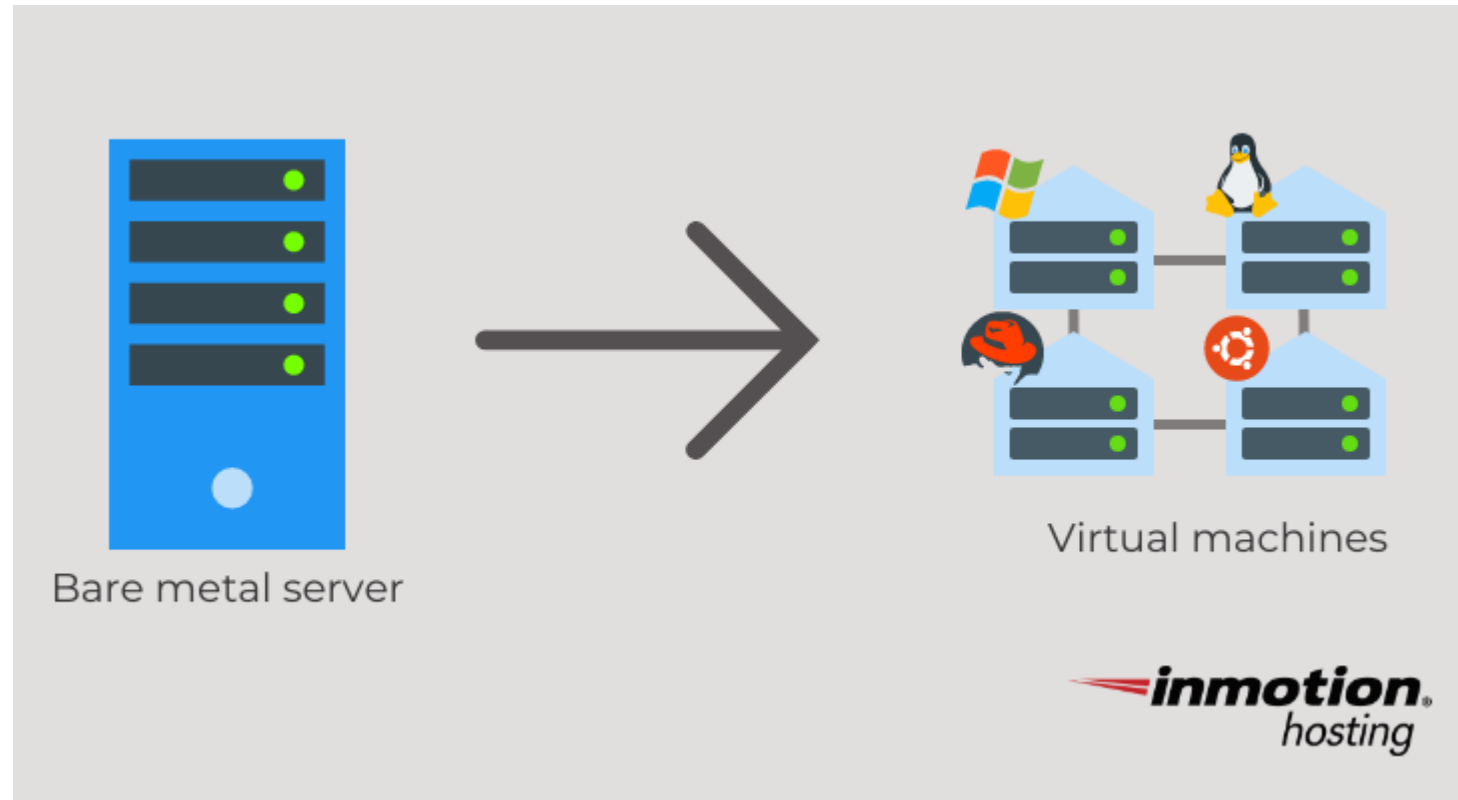
# Not that easy



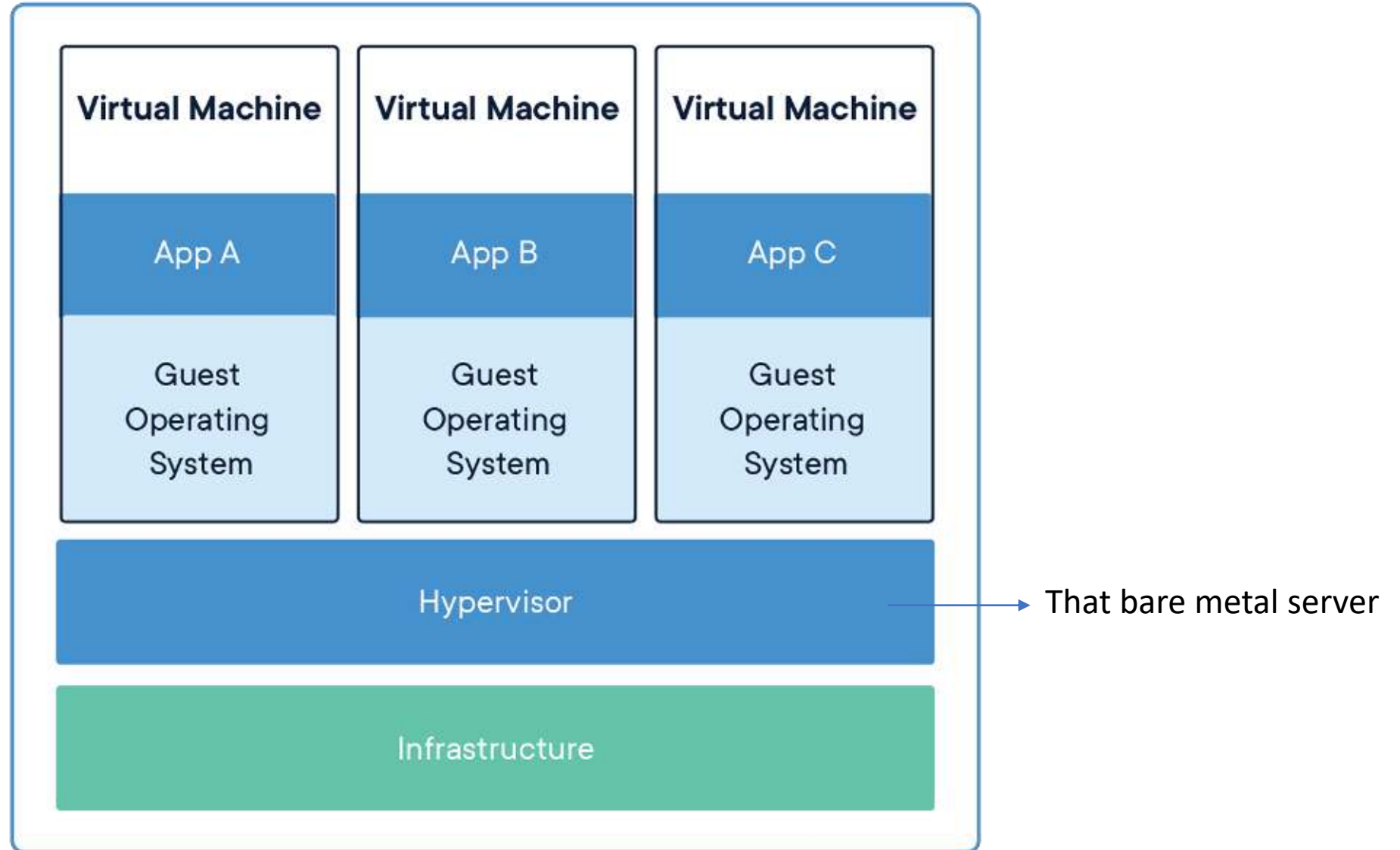
# Create VM Instance (Free VPS)

Google Cloud Platform





A physical server that creates and manages virtual machines is called a hypervisor. A hypervisor can slice (or, “partition”) a portion of drive space and allocate memory so that, in essence, a separate computer can run within a computer.





# VM Instance

## SSH-in-browser

```
Linux bitkubbot 5.10.0-16-cloud-amd64 #1 SMP Debian 5.10.127-1 (2022-06-30) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Jul 11 05:20:10 2022 from 35.235.243.160
sitsakul_portfolio@bitkubbot:~$
```

The image shows the Google Cloud Platform interface for managing VM instances. The top section displays a list of existing instances, and the bottom section shows the 'Create an instance' wizard.

### VM Instances

VM instances are highly configurable virtual machines for running workloads on Google infrastructure. [Learn more](#)

**Filter:** Enter property name or value

Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
<input checked="" type="checkbox"/>	bitkubbot	asia-east1-b	Save \$6 / mo		10.148.0.2 (nic0)	34.126.193.72 (nic0)	SSH
<input type="checkbox"/>	centralized	us-east1-b			10.142.0.3 (nic0)		SSH
<input type="checkbox"/>	instance-1	us-east1-b			10.142.0.2 (nic0)		SSH

### Related actions

- Explore Actifio GO**  
Back up your VMs and setup disaster recovery
- View billing report**  
View and manage your Compute Engine billing
- Monitor VMs**  
View outlier VMs across metrics like CPU and network
- Explore VM logs**  
View, search, analyze, and download VM instance logs
- Set up firewall rules**  
Control traffic to and from a VM instance
- Patch management**  
Schedule patch updates and view patch compliance on VM instances

### Create an instance

To create a VM instance, select one of the options:

- New VM instance**  
Create a single VM instance from scratch
- New VM instance from template**  
Create a single VM instance from an existing template
- New VM instance from machine image**  
Create a single VM instance from an existing machine image
- Marketplace**  
Deploy a ready-to-go solution onto a VM instance

### Machine configuration

**Name:** instance-2

**Labels:** + ADD LABELS

**Region:** us-west4 (Los Angeles) **Zone:** us-west4-b

**Machine family:** GENERAL PURPOSE

**Series:** E2

**Machine type:** e2-medium (2 vCPU, 4 GB memory)

**Machine configuration details:**

Item	Monthly estimate
2 vCPU + 4 GB memory	\$27.55
10 GB balanced persistent disk	\$1.10
Sustained use discount	-\$0.00
<b>Total</b>	<b>\$28.65</b>

**Display device:** Enable to use screen capturing and recording tools





# Running all the time

```
SSH-in-browser
CNO nano S:4 main.py
def crossover(df,a=12,b=26):
    df['EMA_short'] = df['Close'].ewm(span=a, adjust=False,min_periods=a).mean()
    df['EMA_long'] = df['Close'].ewm(span=b, adjust=False,min_periods=b).mean()
    df['trend'] = df.EMA_short > df.EMA_long

    df['trend_shift'] = df.trend.shift(1)
    df.loc[(df.trend == True) & (df.trend_shift == False), 'action'] = 'buy'
    df.loc[(df.trend == False) & (df.trend_shift == True), 'action'] = 'sell'
    return df

def order(df,coin_action,amount=100,order_type='market'):
    if df.iloc[-1]['action'] == 'buy':
        bitkub.place_bid(sym=coin_action, amt=amount, typ=order_type)
        print('Place buy at: ' + str(datetime.now()))
        buy_log.append(['buy', coin, datetime.now()])
        msg = 'Place buy at: ' + str(datetime.now())
        r = requests.post(url, headers=headers, data={'message':msg})
        #print(r.text)

    elif df.iloc[-1]['action'] == 'sell':
        bitkub.place_ask(sym=coin_action, amt=amount/bitkub.ticker(sym=coin_action))
        print('Place sell at: ' + str(datetime.now()))
        sell_log.append(['sell', coin, datetime.now()])
        msg = 'Place sell at: ' + str(datetime.now())
        r = requests.post(url, headers=headers, data={'message':msg})
        #print(r.text)

    else:
        print('no action')
        msg = 'no action'
        r = requests.post(url, headers=headers, data={'message':msg})
        #print(r.text)

def bottrade(coin,int=1,period=30,ema_short=12,ema_long=26,amount=100):
    coin_action = 'THB' + coin
    coin_his = coin+' THB'
    data = get_candle(coin_his,1,30).iloc[:1]
    df = crossover(data,a=ema_short,b=ema_long)
    order(df,coin_action)

    return df
```

