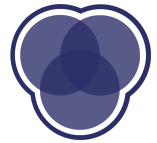
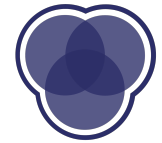
Options Trading Bot with Cloud-Based Infrastructure

Cornell Data Science
Algorithmic Trading



Strategy



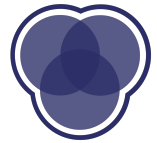
Basic Options

a contract that gives the right, but not the obligation, to buy or sell the underlying asset at a specified price and quantity

strike: the specified price you can buy or sell the equities at

expiration date: date when you can choose to exercise option

premium: the money you pay for the options contract, a fee



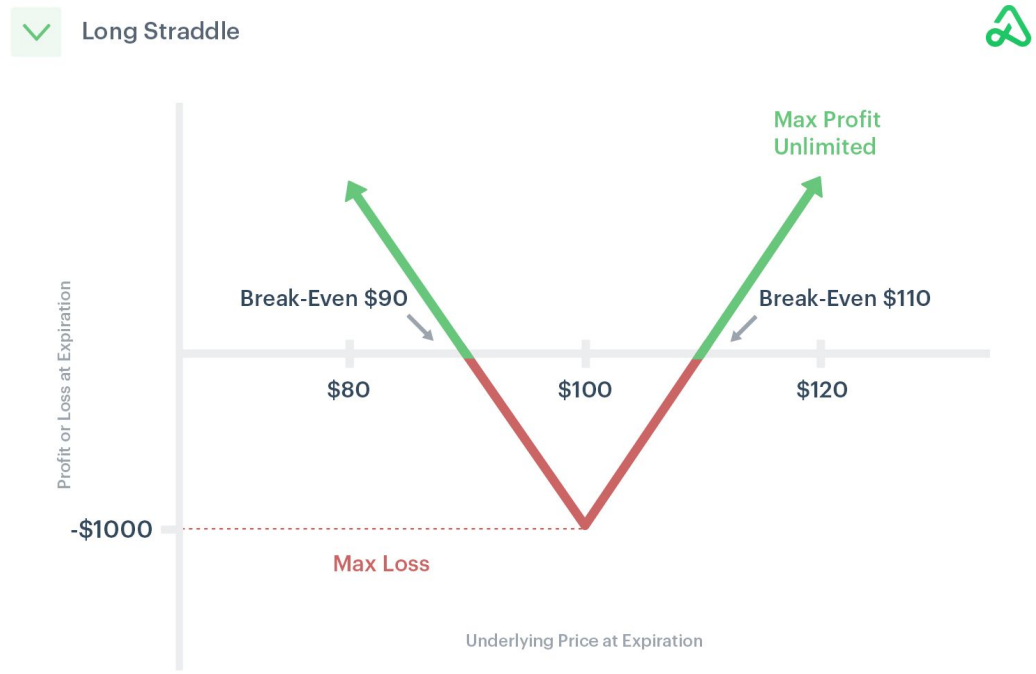
Long Call vs Long Put

Gives the right to buy at strike

Gives the right to sell at strike

Strategy: Long Straddle

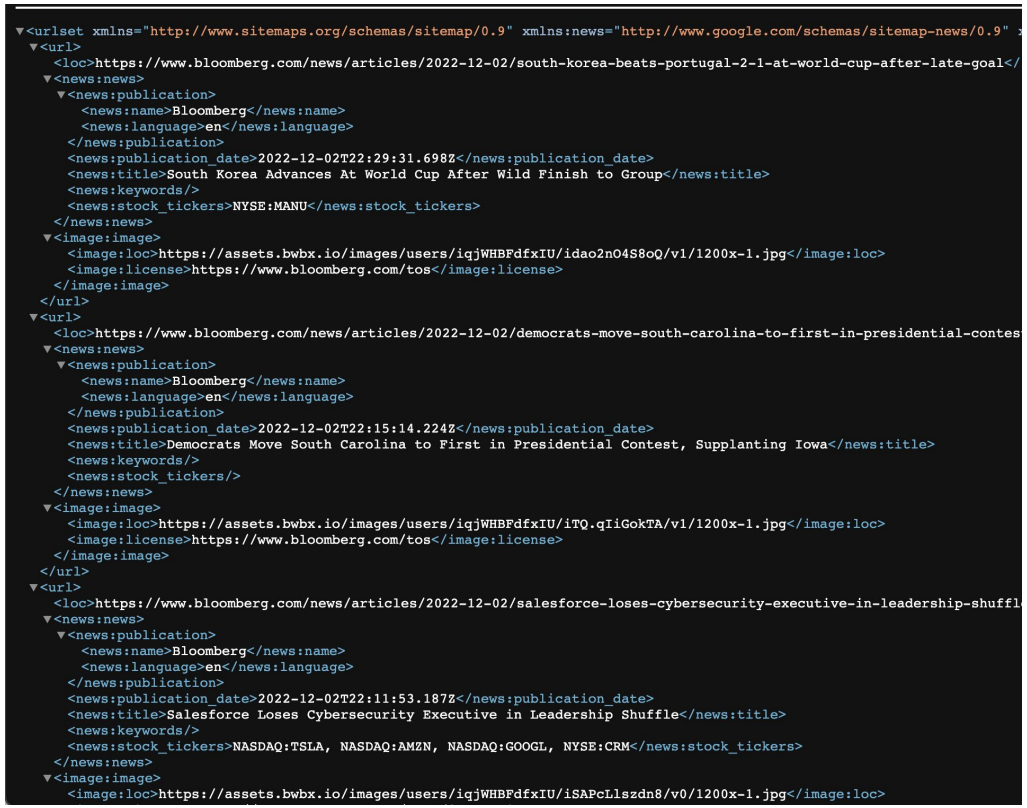
- Buy long call and put at same strike and expiry
- Doesn't matter which direction price swings to
- Idea: capture market reaction to news
- Conjecture: stocks with more intense sentiment will swing more and make it likelier to break-even

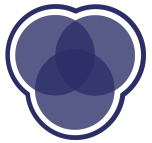


Part I: Scraping

- Wrote code for scraping news utilizing mainly Bloomberg for article source
- Utilized Beautifulsoup module for parsing
- Thought needed for type of RSS feeds used
- Outputted dictionary with tickers as key and the value as tuple with published date and article headline

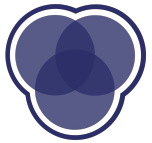
```
'ADM': [('Grimy Ships for Iron Ore Are Now Being Used to\xa0Carry Food Crops to Asia',
  '2022-12-02T00:00:00.000Z'),
 ('Biden Proposes Overhaul of US Biofuel Law to Boost EV Makers Like Tesla',
  '2022-12-01T17:02:34.198Z'),
 ('Biden Set to Raise Refiners' Biofuels Quotas in Green Push',
  '2022-11-30T23:27:17.117Z')],
'VSAT': [('SpaceX Wins FCC Approval to Launch 7,500 Starlink Satellites',
  '2022-12-01T22:46:37.370Z')],
'TSN': [('Tyson CFO Pleads Not Guilty to Public Intoxication, Trespassing',
  '2022-12-01T22:45:54.819Z')],
'WFC': [('Your Evening Briefing: The High Price of High Interest Rates',
  '2022-12-01T22:28:29.051Z'),
 ('Kroger Raises Profit Outlook as Regulators Weigh Albertsons Purchase',
  '2022-12-01T20:43:53.867Z'),
 ('Wells Fargo Cuts Hundreds More Mortgage Employees on Industry Slowdown',
  '2022-12-01T17:54:47.449Z')],
'AMCX': [('AMC Networks Says Restructuring Charges Could Hit $475 Million',
  '2022-12-01T22:26:20.815Z')],
```





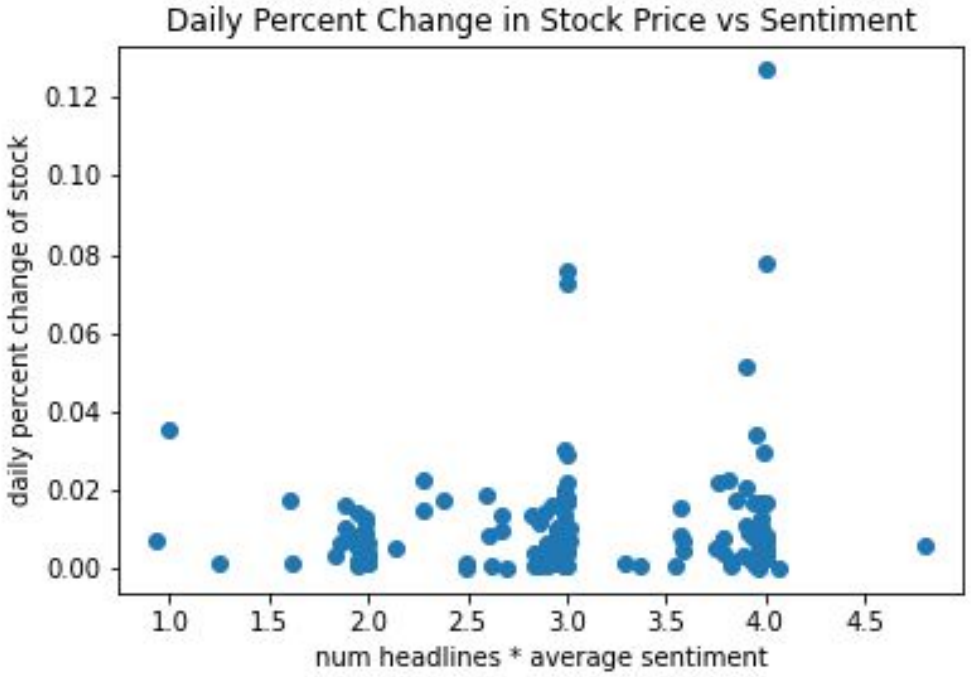
Part II: Sentiment Analysis

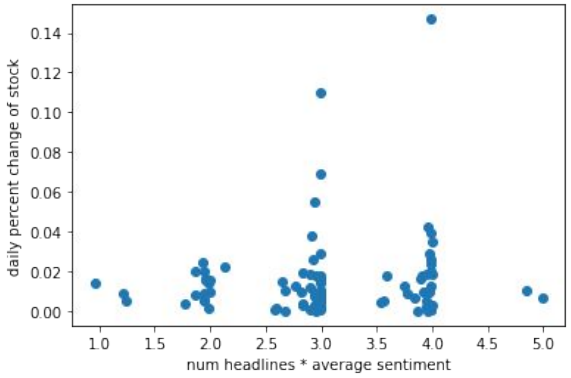
- Trial and error with many pretrained sentiment models
  - Vader: social media posts oriented
  - TextBlob: didn't agree with the numbers
- Eventually decided on Hugging Face's model
- Decision on whether it has extreme sentiment or not
  - Number of headlines: affect audience reach
  - Sentiment score output of the model
  - Take the product to interrelate two values (sum doesn't provide as much scaling, ends up same as just taking number of headlines)

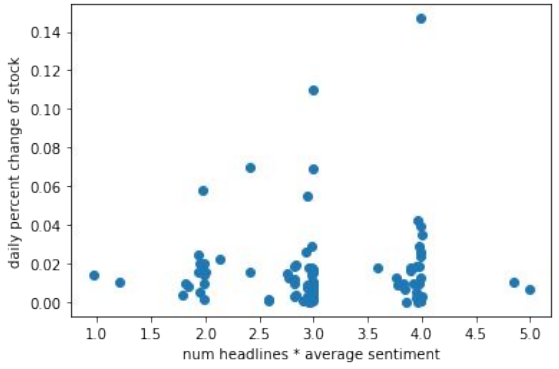


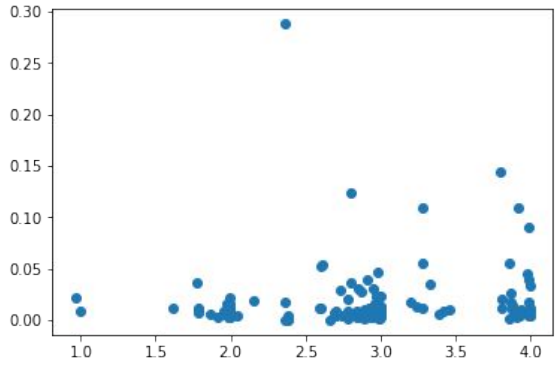
Backtest Result

- Hypothesis somewhat correct
- Higher products have higher "spreads" of daily price change, implying greater chance of greater movements in prices









Part III: Buy

- Pulled live options data from yfinance
- yfinance only produce either a list of calls or puts
  - paired calls and puts with the same expiration dates and strike prices to create a straddle
- Picked the straddle (call and put pair) with closest expiration to make it easier to keep track of the results
- Chose a straddle with minimum premium

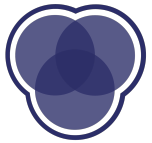
```python
def getOptions(ticker):
    tick = yf.Ticker(ticker)
    try:
        expirations = tick.options
    except Exception as e:
        logger.info('Issue fetching options. No option or not an American ticker. Skipping...')
        return pd.DataFrame()

    if len(expirations) == 0:
        # no options error also
        return pd.DataFrame()

    puts = pd.DataFrame()
    for date in expirations[:1]:  # take only earliest expiration
        opt = tick.option_chain(date)
        opt = pd.DataFrame().append(opt.puts)
        opt["ExpirationDate"] = date
        puts = puts.append(opt)

    calls = pd.DataFrame()
    for date in expirations[:1]:  # take only earliest expiration
        opt = tick.option_chain(date)
        opt = pd.DataFrame().append(opt.calls)
        opt["ExpirationDate"] = date
        calls = calls.append(opt)

    options_full = calls.merge(puts, on=["strike", "ExpirationDate"])
    return options_full
```

```python
def strategy_handler(event, context):

    company_data = get_stock_sentiment()
    for company in company_data:
        if not company_data[company][0] > 0:
            continue

        logger.info(f'Getting options for company {company}')
        options = getOptions(company)
        if not options.empty:
            logger.info('Found options for the company')
            # smallest difference in last price
            min_diff = abs(options["lastPrice_x"] - options["lastPrice_y"]).min()
            bought = options.loc[abs(
                options["lastPrice_x"] - options["lastPrice_y"]) == min_diff]
            if bought.empty:
                continue
            bought = bought.drop(columns=['bid_x', 'ask_x', 'change_x', 'percentChange_x',
                                          'volume_x', 'openInterest_x',
                                          'inTheMoney_x', 'contractSize_x', 'currency_x',
                                          'lastTradeDate_x', 'impliedVolatility_x',
                                          'bid_y', 'ask_y', 'change_y', 'percentChange_y',
                                          'volume_y', 'openInterest_y',
                                          'inTheMoney_y', 'contractSize_y', 'currency_y',
                                          'lastTradeDate_y', 'impliedVolatility_y'])

            company = company.split('/')
            company = '.'.join(company)
            currentTimeInMillis = (int)(time.time()) * 1000

            call_uuid = str(uuid.uuid4())
            put_uuid = str(uuid.uuid4())
```

Results: Options Bought

(bought earlier this week with most expiry this Fri)

index	stockName	putOrCall	contract_id	strike	premiumFee	expiryDate
0	GS	Call	GS221202C00382500	382.5	3.44	2022-12-02
1	GS	Put	GS221202P00382500	382.5	4.3	2022-12-02
2	RY	Call	RY221216C00100000	100.0	1.35	2022-12-16
3	RY	Put	RY221216P00100000	100.0	2.55	2022-12-16
4	JPM	Call	JPM221202C00137000	137.0	1.18	2022-12-02
5	JPM	Put	JPM221202P00137000	137.0	1.64	2022-12-02
6	AMZN	Call	AMZN221202C00092000	92.0	1.95	2022-12-02
7	AMZN	Put	AMZN221202P00092000	92.0	1.5	2022-12-02
8	MS	Call	MS221202C00091000	91.0	1.07	2022-12-02
9	MS	Put	MS221202P00091000	91.0	0.95	2022-12-02
10	GOOGL	Call	GOOGL221202C00095000	95.0	1.45	2022-12-02
11	GOOGL	Put	GOOGL221202P00095000	95.0	1.19	2022-12-02
12	NYT	Call	NYT221216C00035000	35.0	1.1	2022-12-16
13	NYT	Put	NYT221216P00035000	35.0	0.91	2022-12-16
14	META	Call	META221202C00110000	110.0	1.73	2022-12-02
15	META	Put	META221202P00110000	110.0	2.2	2022-12-02
16	MCO	Call	MCO221216C00300000	300.0	5.4	2022-12-16
17	MCO	Put	MCO221216P00300000	300.0	9.2	2022-12-16
18	STLA	Call	STLA221216C00015000	15.0	0.6	2022-12-16
19	STLA	Put	STLA221216P00015000	15.0	0.37	2022-12-16
20	AIR	Call	AIR221216C00045000	45.0	1.88	2022-12-16
21	AIR	Put	AIR221216P00045000	45.0	2.0	2022-12-16
22	BA	Call	BA221202C00175000	175.0	2.97	2022-12-02
23	BA	Put	BA221202P00175000	175.0	2.6	2022-12-02
24	WMT	Call	WMT221202C00152500	152.5	1.45	2022-12-02
25	WMT	Put	WMT221202P00152500	152.5	0.95	2022-12-02
26	C	Call	C221202C00047500	47.5	0.62	2022-12-02
27	C	Put	C221202P00047500	47.5	0.5	2022-12-02
28	BAC	Call	BAC221202C00037000	37.0	0.32	2022-12-02
29	BAC	Put	BAC221202P00037000	37.0	0.48	2022-12-02
30	GM	Call	GM221202C00039500	39.5	0.76	2022-12-02
31	GM	Put	GM221202P00039500	39.5	0.48	2022-12-02
32	AAPL	Call	AAPL221202C00141000	141.0	2.22	2022-12-02
33	AAPL	Put	AAPL221202P00141000	141.0	1.93	2022-12-02
34	DIS	Call	DIS221202C00095000	95.0	1.24	2022-12-02
35	DIS	Put	DIS221202P00095000	95.0	1.46	2022-12-02
36	NFLX	Call	NFLX221202C00280000	280.0	5.52	2022-12-02
37	NFLX	Put	NFLX221202P00280000	280.0	4.35	2022-12-02
38	SHEL	Call	SHEL221216C00057500	57.5	1.55	2022-12-16
39	SHEL	Put	SHEL221216P00057500	57.5	1.3	2022-12-16
40	HOOD	Call	HOOD221202C00009000	9.0	0.32	2022-12-02
41	HOOD	Put	HOOD221202P00009000	9.0	0.17	2022-12-02
42	KR	Call	KR221202C00049000	49.0	1.92	2022-12-02
43	KR	Put	KR221202P00049000	49.0	1.59	2022-12-02
44	CVX	Call	CVX221202C00180000	180.0	2.77	2022-12-02
45	CVX	Put	CVX221202P00180000	180.0	1.63	2022-12-02
46	CM	Call	CM221216C00047500	47.5	1.28	2022-12-16
47	CM	Put	CM221216P00047500	47.5	1.25	2022-12-16
48	RACE	Call	RACE221202C00217500	217.5	3.3	2022-12-02
49	RACE	Put	RACE221202P00217500	217.5	2.15	2022-12-02
50	TSLA	Call	TSLA221202C00180000	180.0	5.23	2022-12-02
51	TSLA	Put	TSLA221202P00180000	180.0	4.28	2022-12-02
52	EZJ	Call	EZJ230120C00027000	27.0	1.3	2023-01-20
53	EZJ	Put	EZJ230120P00027000	27.0	1.75	2023-01-20
54	MSFT	Call	MSFT221202C00240000	240.0	3.3	2022-12-02
55	MSFT	Put	MSFT221202P00240000	240.0	2.85	2022-12-02
56	BLK	Call	BLK221202C00715000	715.0	9.0	2022-12-02
57	BLK	Put	BLK221202P00715000	715.0	10.0	2022-12-02
58	INTC	Call	INTC221202C00029000	29.0	0.36	2022-12-02
59	INTC	Put	INTC221202P00029000	29.0	0.44	2022-12-02
60	HPQ	Call	HPQ221202C00029000	29.0	0.35	2022-12-02
61	HPQ	Put	HPQ221202P00029000	29.0	0.46	2022-12-02



Results: Returns

Exercised 12/2

Total Premium = $10057

Total Profit = $16268

Net Profit = $6211

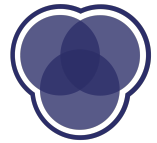
61.76% return!!

*NFLX, META, TSLA, MSFT all spiked this week

	Stock	Strike	Premium	Type	Profit	Net	Exercised Price
0	GS	382.5	7.74	Put	4.100006	-3.639994	378.399994
1	JPM	137.0	2.82	Put	3.660004	0.840004	133.339996
2	AMZN	92.0	3.45	Call	3.360001	-0.089999	95.360001
3	MS	91.0	2.02	Call	1.660004	-0.359996	92.660004
4	GOOGL	95.0	2.64	Call	5.769997	3.129997	100.769997
5	META	110.0	3.93	Call	14.040001	10.110001	124.040001
6	BA	175.0	5.57	Call	8.449997	2.879997	183.449997
7	WMT	152.5	2.40	Put	1.020004	-1.379996	151.479996
8	C	47.5	1.12	Put	0.689999	-0.430001	46.810001
9	BAC	37.0	0.80	Put	1.189999	0.389999	35.810001
10	GM	39.5	1.24	Call	0.639999	-0.600001	40.139999
11	AAPL	141.0	4.15	Call	7.000000	2.850000	148.000000
12	DIS	95.0	2.70	Call	4.690002	1.990002	99.690002
13	NFLX	280.0	9.87	Call	41.989990	32.119990	321.989990
14	HOOD	9.0	0.49	Call	1.050000	0.560000	10.050000
15	KR	49.0	3.51	Put	1.720001	-1.789999	47.279999
16	CVX	180.0	4.40	Call	3.630005	-0.769995	183.630005
17	RACE	217.5	5.45	Call	9.179993	3.729993	226.679993
18	TSLA	180.0	9.51	Call	16.250000	6.740000	196.250000
19	MSFT	240.0	6.15	Call	16.059998	9.909998	256.059998
20	BLK	715.0	19.00	Put	15.299988	-3.700012	699.700012
21	INTC	29.0	0.80	Call	0.430000	-0.370000	29.430000
22	HPQ	29.0	0.81	Call	0.799999	-0.010001	29.799999

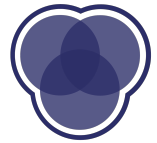
Infrastructure Team

Jack, Uday, Rohan, Rucha



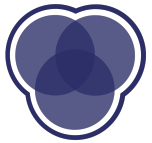
Goal

Create a scalable and reliable cloud-based infrastructure for the trading bot



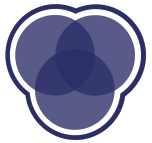
Why use the Cloud?

1. Automated Infrastructure
    - Fully managed servers / databases
    - AWS CDK/Infrastructure as Code (IaC) - spin up everything with one command.
2. Scalability
    - AWS Lambda/EC2 provide autoscaling (i.e. more instances are automatically created to handle higher load).
3. On-Demand Resources
    - Save money by not having extra instances run all the time
    - Scale up/down when you need to

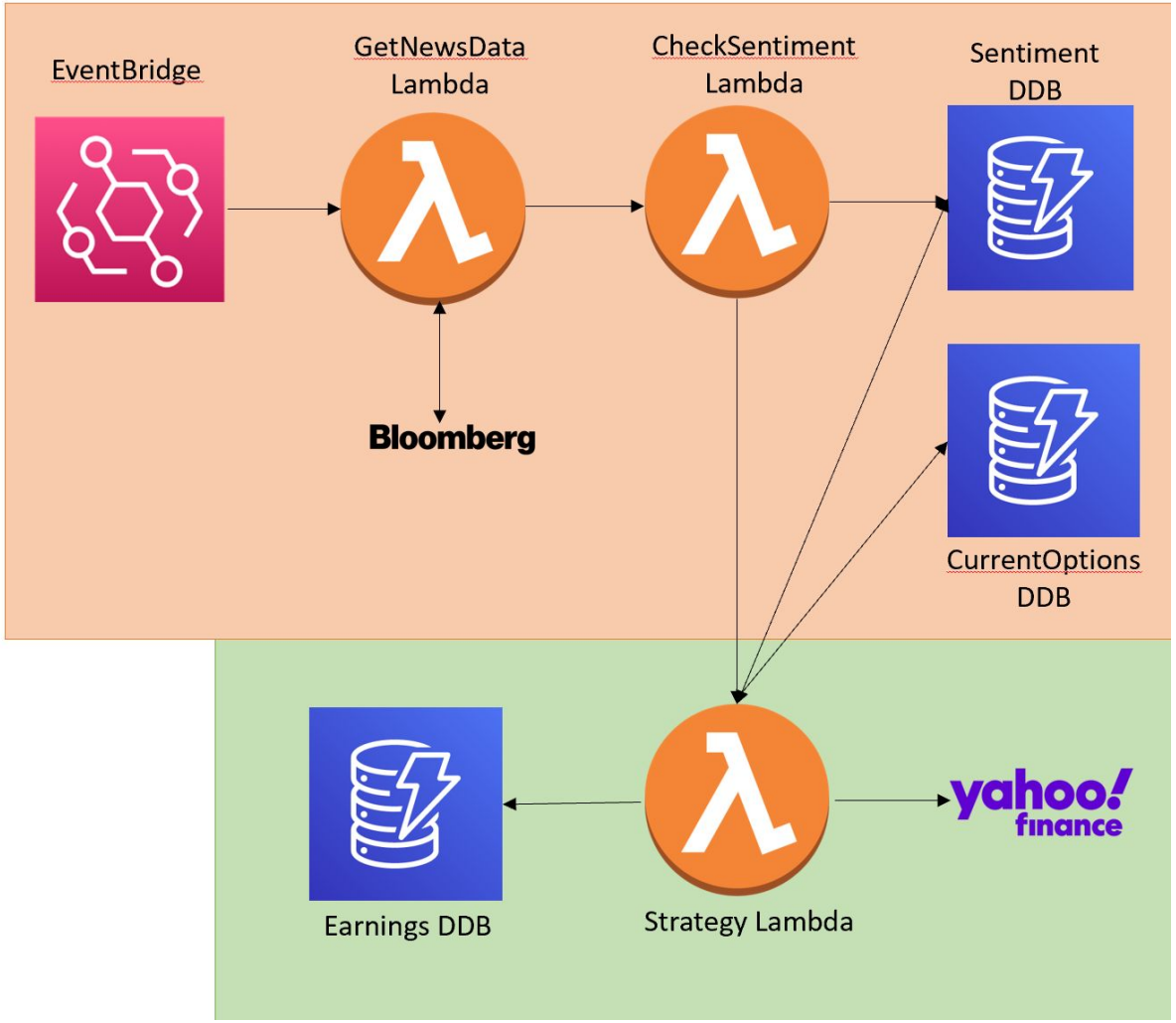


Technologies/AWS Services Used

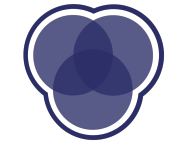
- Amazon Web Services - cloud service provider
- AWS Lambda
- Docker
- AWS DynamoDB
- AWS Step Functions
- AWS CDK



Trading Bot System Architecture



Demo

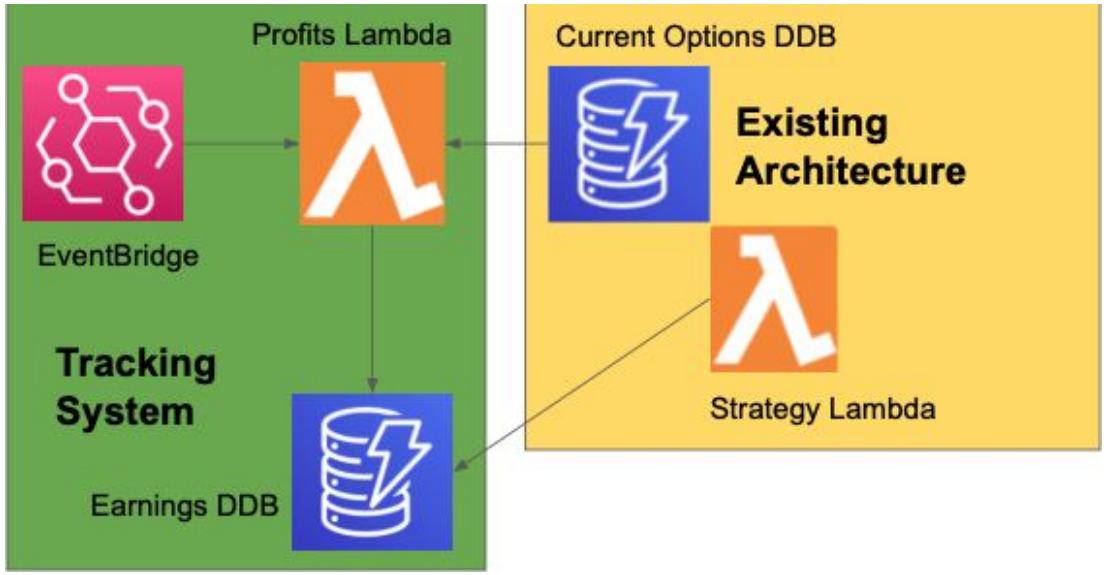


Calculating Profit and Tracking Money

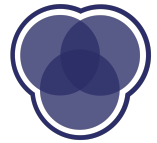
1. Separate “Earnings” database that acts as a ledger for all option transactions, and stores the current balance

2. With each transaction (e.g. call/put contract purchase), the balance is updated

3. At the end of each market day, an event bridge calls a lambda function to find expiring contracts, calculate the profit (if applicable), and record it in our ledger

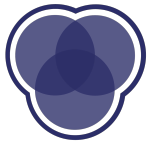


Back to the Demo



Future Work/Possible Extensions

1. Real time testing
2. Alternative trading strategies which could fit in our already defined architecture and sentiment analysis
3. Scale up cloud infrastructure to handle more heavy or more frequent requests to execute trades
    a. Data streams
    b. Concurrent trades
4. Using alternative APIs
    a. Weren't able to get IBKR (Interactive Brokers) to work initially, so had to pivot



Thank you!