

***Docker-compose,  
vagrant і інші  
«страшні»  
слова***



***[bit.ly/msp-docker](https://bit.ly/msp-docker)***

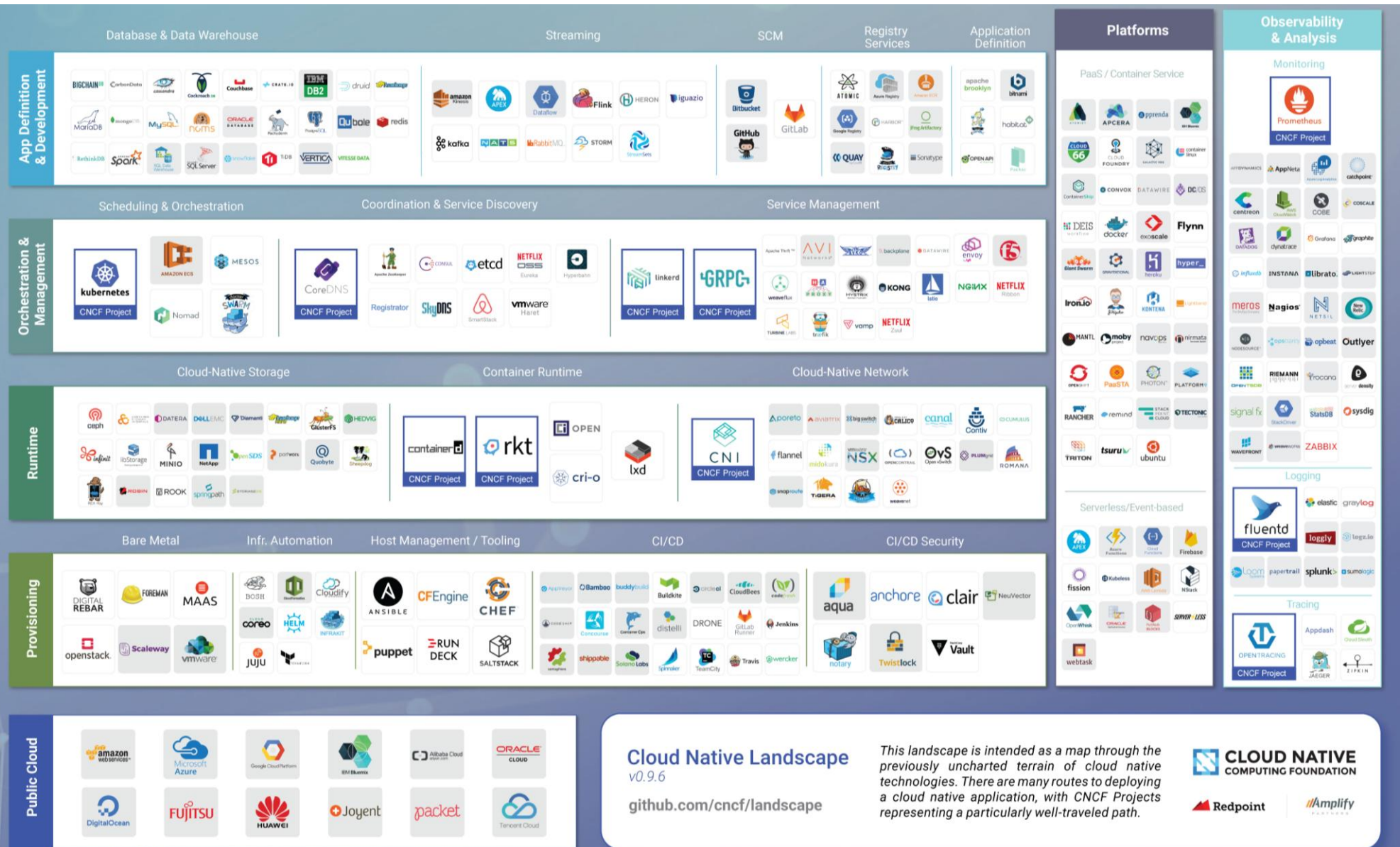
# About me

- Work as DevOps Engineer at PDFfiller
- IT KPI (co-founder)
- Linux & Open Source enthusiast
- In love with JavaScript & Python
- Write on Bash and PHP



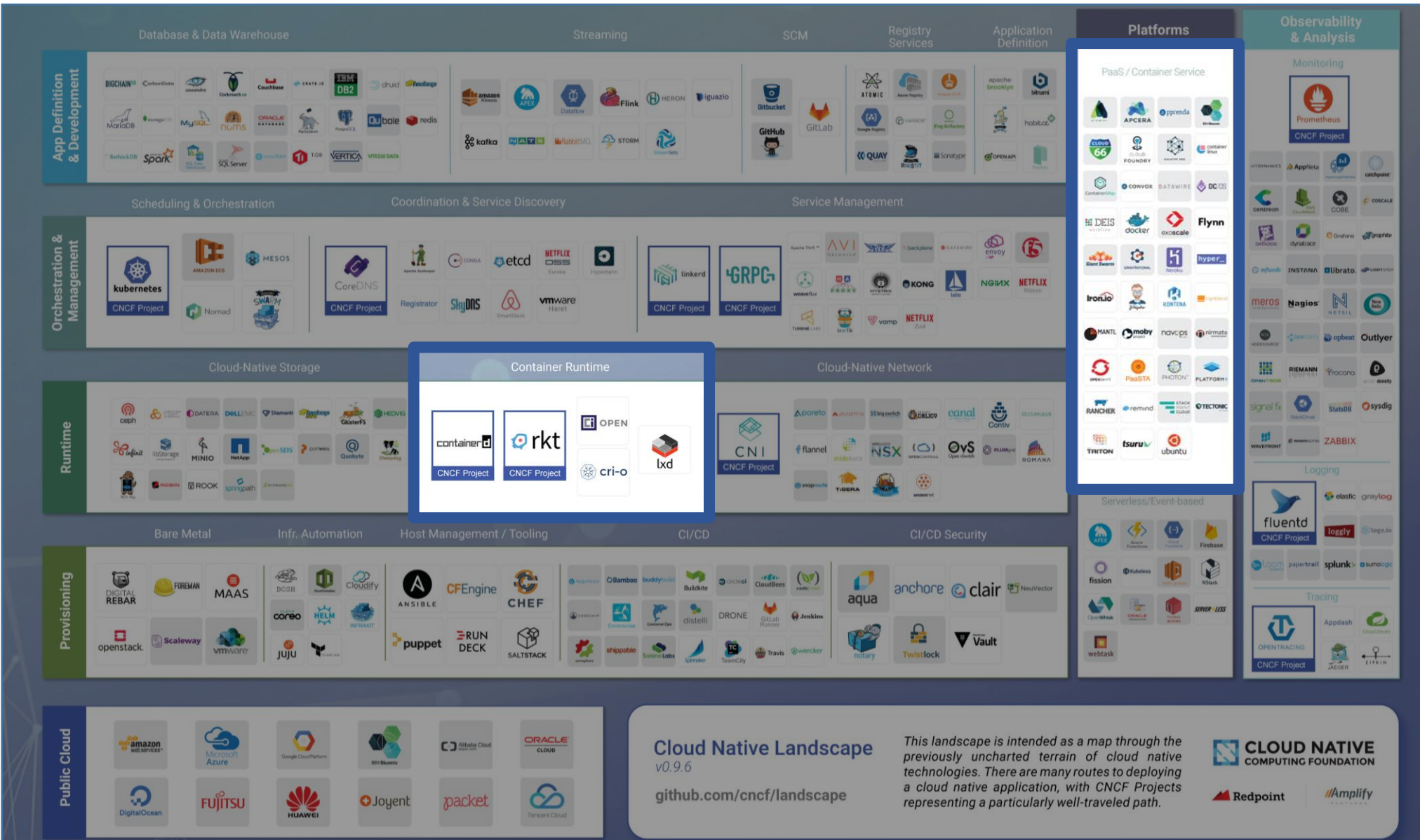


# So many...



This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.

# So many...



This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.

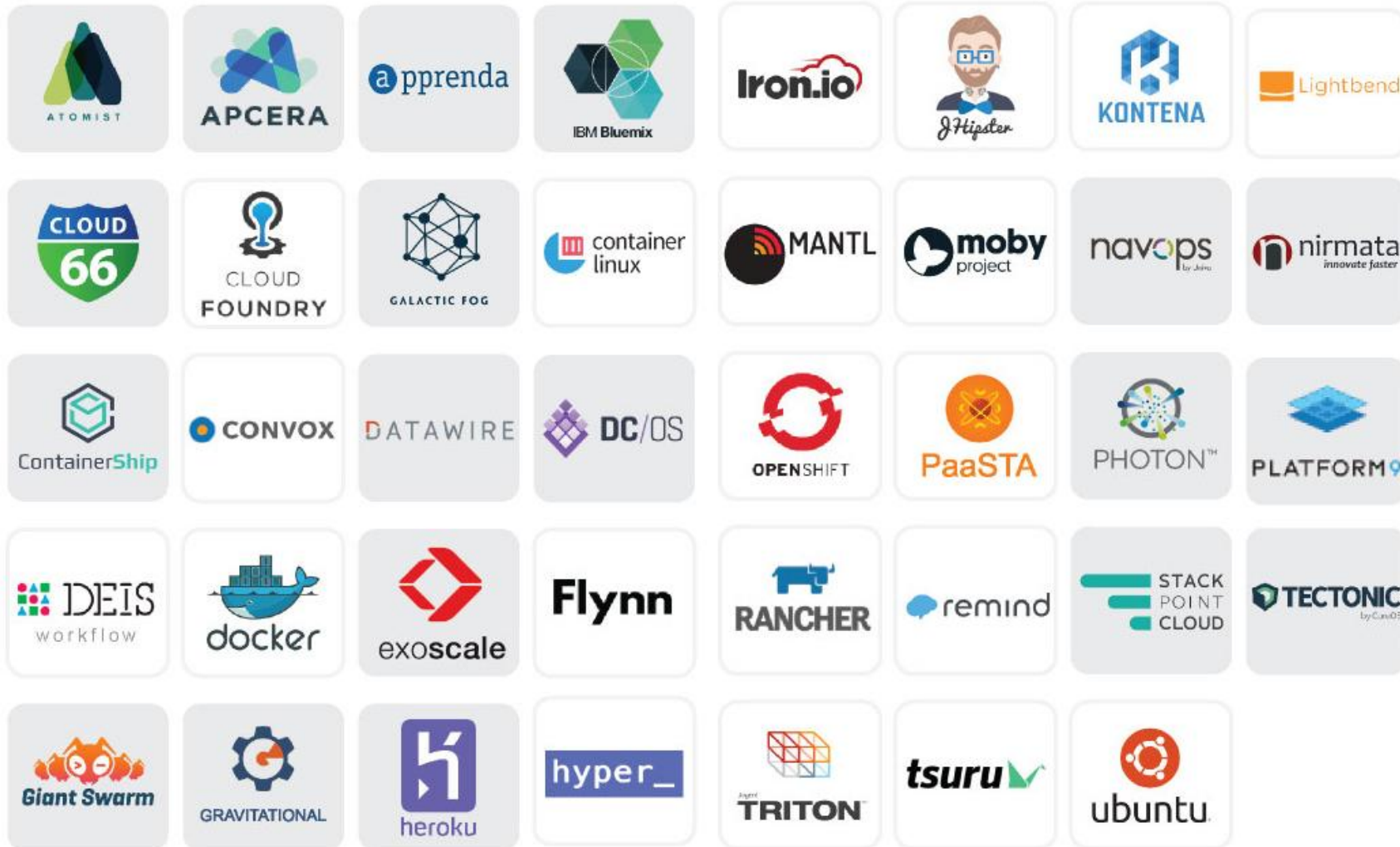


# Container runtime



# Container services & PaaS

PaaS / Container Service



# Just container services, which can run local

Container Service



The background is a solid dark blue color. Overlaid on this background is a complex, abstract pattern of thin, light blue lines. These lines form a network of interconnected paths, resembling a circuit board or a data flow diagram. The lines vary in thickness and direction, creating a sense of depth and movement. Some lines are straight, while others are curved or bent at right angles. The overall effect is a high-tech, digital aesthetic.

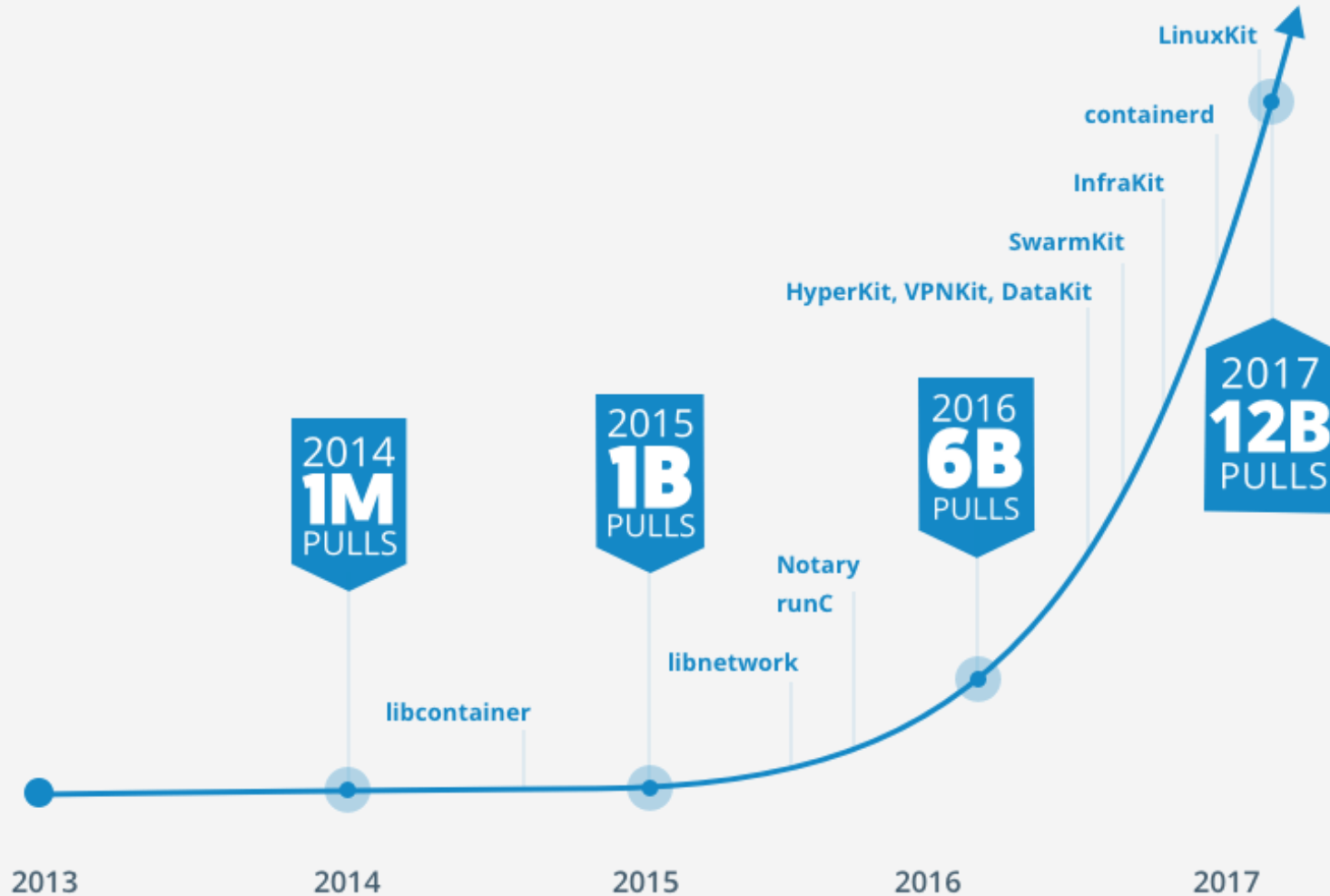
*Docker*



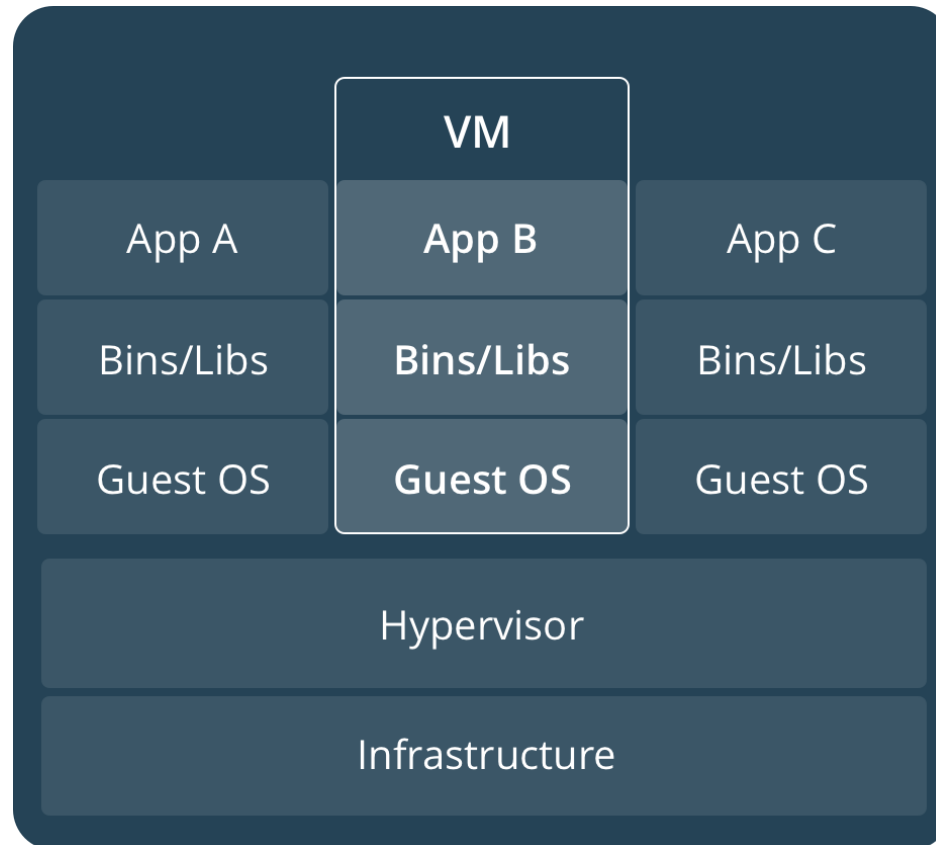
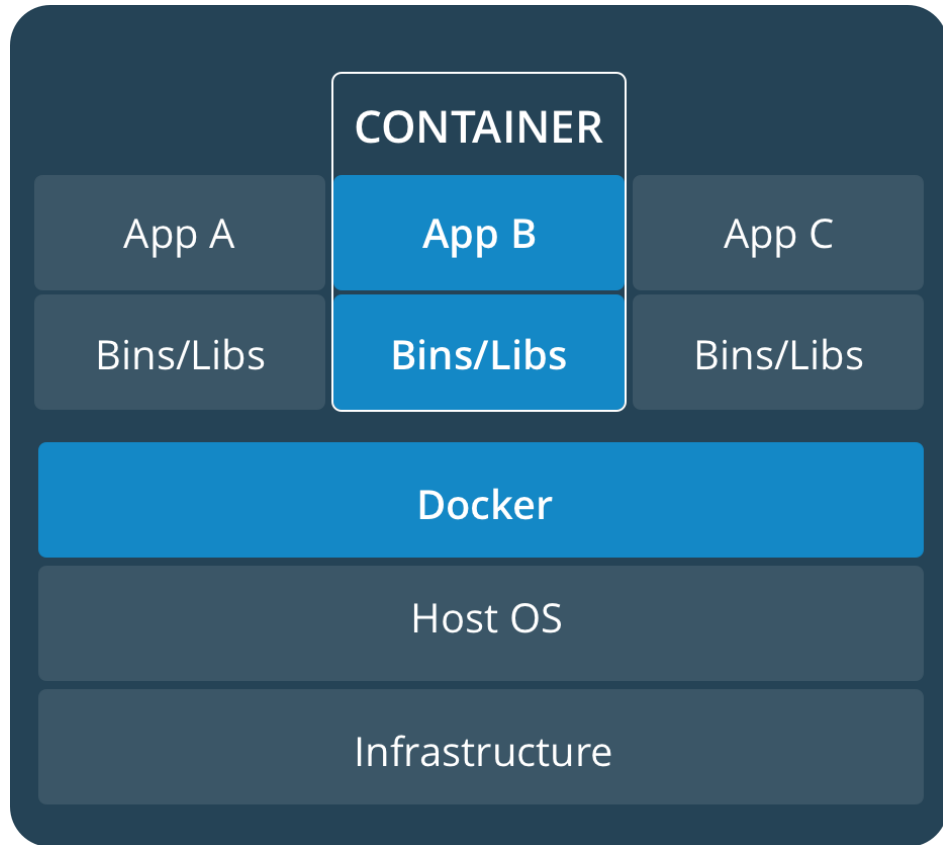
# Docker Era

## Pulls

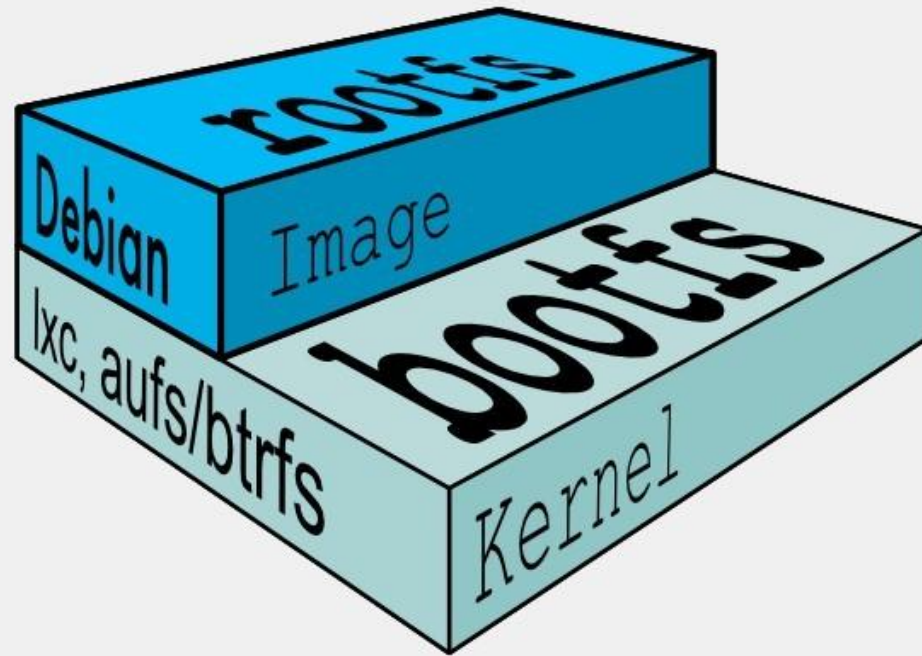
12,000,000,000  
11,000,000,000  
10,000,000,000  
9,000,000,000  
8,000,000,000  
7,000,000,000  
6,000,000,000  
5,000,000,000  
4,000,000,000  
3,000,000,000  
2,000,000,000  
1,000,000,000



# Container vs VM

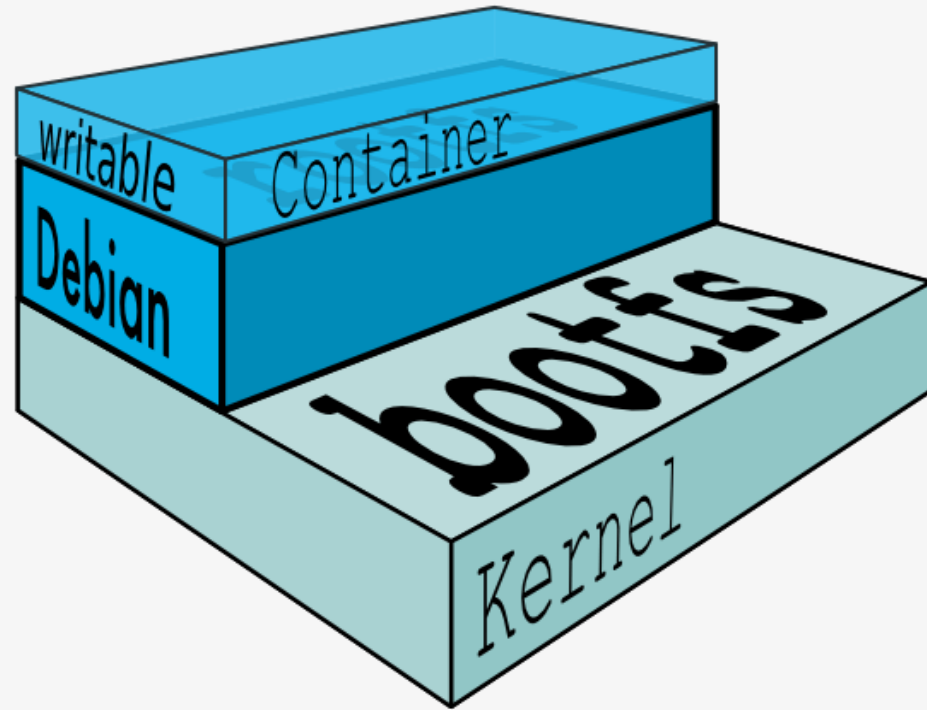


# How it works

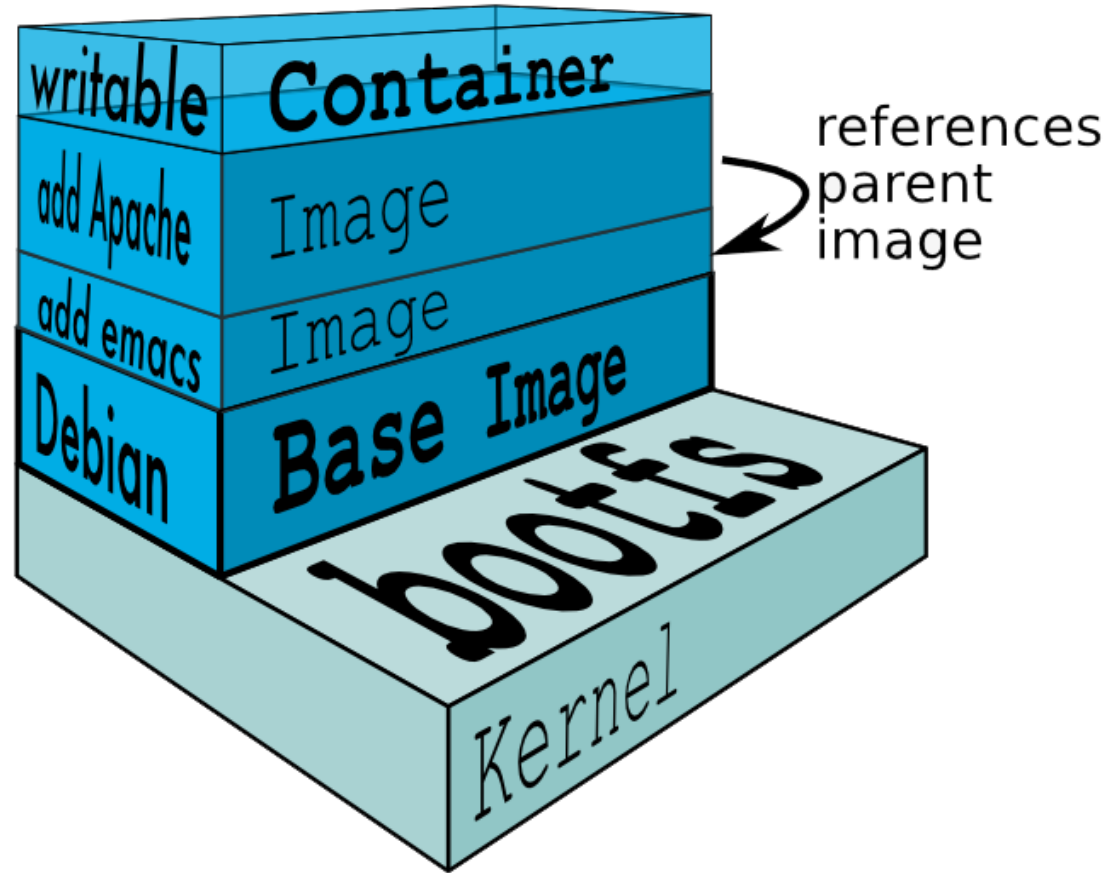




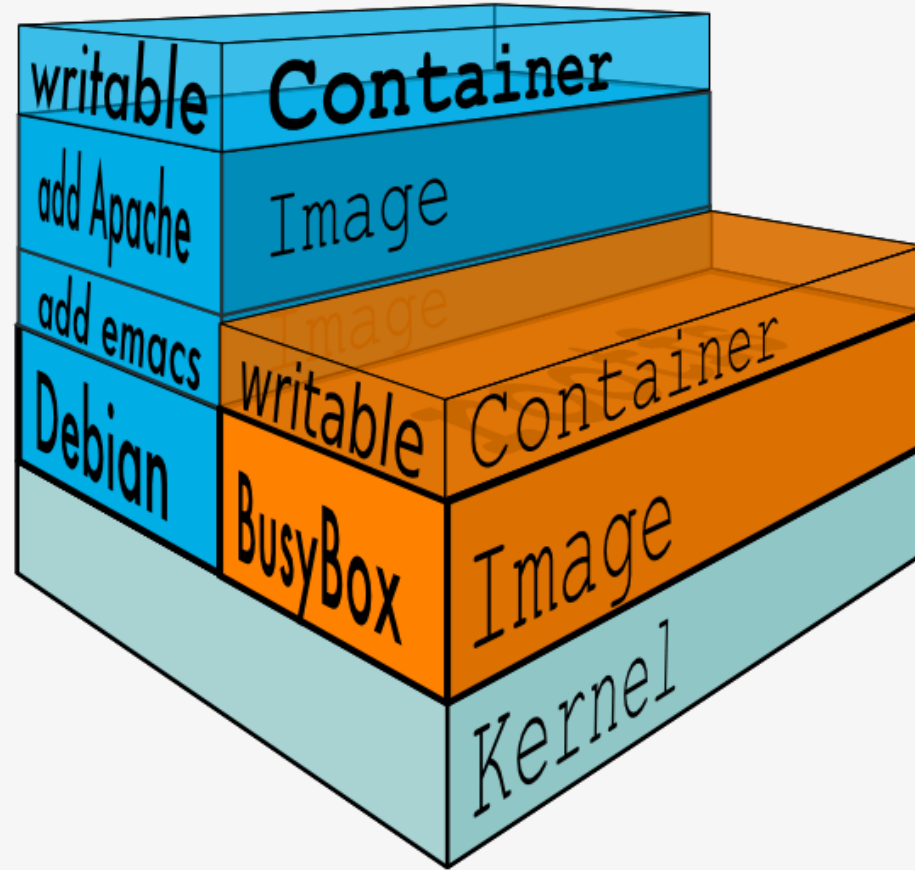
# How it works



# How it works



# How it works





# Dockerfile



# Dockerfile

```
1 FROM eбораas/apache:latest
2
3 LABEL maintainer="MaxymVlasov <m@il.com>"
4
5 # Add rewrite module to Apache
6 # Need for port forwarding for NodeJS
7 RUN a2enmod rewrite && service apache2 restart
8
9 # Change default Apache configuration
10 COPY 000-default.conf /etc/apache2/sites-available/000-default.conf
11 COPY ports.conf /etc/apache2/ports.conf
12
13 # Open port
14 EXPOSE 8080
15
16 # Run Apache server in background
17 CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```

# Dockerfile

```
1  ARG VERSION=latest
2  FROM busybox:${VERSION}
3
4
5  ARG VERSION
6
7  RUN echo ${VERSION} > version
```



# Dockerfile

```
1  ARG VERSION=latest
2  FROM busybox:${VERSION}
3
4  ENV VERSION previous
5  ARG VERSION
6
7  RUN echo ${VERSION} > version
```

# Dockerfile

```
1  ARG VERSION=latest
2  FROM busybox:${VERSION}
3
4  ARG VERSION
5  ENV VERSION=${VERSION:-0}
6
7  RUN echo ${VERSION} > version
```

# Dockerfile

```
1  ARG VERSION=latest
2  FROM busybox:${VERSION}
3
4  ARG VERSION
5  ENV VERSION=${VERSION:-0}
6
7  RUN echo ${VERSION} > version
8
9  ADD http://bit.ly/raw or binary file /app/file
10 COPY . /app
```



# Dockerfile

```
1  FROM busybox:latest
2
3  WORKDIR /a
4  WORKDIR b
5  WORKDIR c
6
7  RUN pwd # Output: /a/b/c
```



# Dockerfile

```
1  FROM busybox:latest
2
3  WORKDIR /a
4  WORKDIR b
5  WORKDIR c
6
7  RUN pwd # Output: /a/b/c
8
9  ENTRYPOINT ["ping"]
10 CMD ["8.8.8.8"]
```

# Dockerfile

```
1 FROM busybox:latest
2
3 LABEL maintainer="Maksym Vlasov <m@il.com>"
```

# Dockerfile

```
1 FROM busybox:latest
2
3 LABEL maintainer="Maksym Vlasov <m@il.com>"
4
5 EXPOSE 9000 9001
```



# Dockerfile

```
1 FROM busybox:latest
2
3 LABEL maintainer="Maksym Vlasov <m@il.com>"
4
5 EXPOSE 9000 9001
6
7 VOLUME ["/data"]
```

# Dockerfile

```
1  FROM busybox:latest
2
3  LABEL maintainer="Maksym Vlasov <m@il.com>"
4
5  EXPOSE 9000 9001
6
7  VOLUME ["/data"]
8
9  HEALTHCHECK --interval=30s \
10             --timeout=30s \
11             --start-period=0s \
12             --retries=3 \
13  CMD wget http://localhost/ || exit 1
```

# Dockerfile

---

```
1 FROM busybox:latest
2
3 SHELL ["/bin/sh", "-c"]
```

# Dockerfile

---

```
1 FROM busybox:latest
2
3 SHELL ["/bin/sh", "-c"]
4
5 RUN addgroup group && \
6     adduser -D -H user -G group
7
8 USER user:group
```



# Dockerfile

---

```
1  FROM busybox:latest
2
3  SHELL ["/bin/sh", "-c"]
4
5  RUN addgroup group && \
6      adduser -D -H user -G group
7
8  USER user:group
9
10 STOPSIGNAL 9 # SIGKILL
```

# Dockerfile

---

```
1  FROM busybox:latest
2
3  SHELL ["/bin/sh", "-c"]
4
5  RUN addgroup group && \
6      adduser -D -H user -G group
7
8  USER user:group
9
10 STOPSIGNAL 9 # SIGKILL
11
12 ONBUILD ADD . /app/src
13 ONBUILD RUN echo "based on busybox"
```

The background is a dark blue gradient with a complex, glowing circuit pattern. The pattern consists of numerous thin, white lines that form a dense network of interconnected paths, resembling a printed circuit board or a neural network. Some of these lines are thicker and more prominent, while others are thin and delicate. Scattered throughout the circuit are small, bright yellow-green dots, which appear to be light sources or data points within the network. The overall effect is a high-tech, digital aesthetic.

# *Docker-compose*

# Docker-compose.yml





# Docker-compose.yml

```
1  version: '3'
2
3  services:
4    apache:
5      image: maxymvlasov/apache-pdfiller
6      ports:
7        - "8080"
8      volumes:
9        - ./html:/var/www/
10     environment:
11       - VIRTUAL_HOST=0.0.0.0
12       - VIRTUAL_PROTO=http
13   nodejs:
14     image: maxymvlasov/nodejs-pdfiller
15     ports:
16       - "8000:8000"
17     environment:
18       - "constraint:node==node-1"
19   nginx:
20     image: maxymvlasov/nginx-pdfiller
21     ports:
22       - "80:80"
23     volumes:
24       - /var/run/docker.sock:/tmp/docker.sock:ro
25     environment:
26       - VIRTUAL_PORT=8080
27       - "constraint:node==node-1"
```

# Docker-compose.yml

```
1  version: '3'
2
3  services:
4    consul:
5      build: ../dev_docker_compose/consul
6      network_mode: host
7      restart: always
8      volumes:
9        - consuldb:/consul/data
10
11  volumes:
12    consuldb:
13      external: true
```

The background is a solid dark blue color. Overlaid on this are intricate, light blue circuit-like patterns that resemble a complex network or a stylized map. These patterns consist of numerous thin lines that branch out and connect at various points. At some of these connection points, there are small, glowing yellow-green dots, giving the impression of active nodes or data points within the network. The overall aesthetic is technological and futuristic.

*Vagga*

# vagga.yaml

1 containers:



# vagga.yaml

```
1 containers:  
2   ubuntu-consul:
```





# vagga.yaml

```
1  containers:
2    ubuntu-consul:
3      setup:
4        - !Ubuntu xenial
5        - !Install [unzip, wget, ca-certificates]
```



# vagga.yaml

```
1  containers:
2    ubuntu-consul:
3      setup:
4        - !Ubuntu xenial
5        - !Install [unzip, wget, ca-certificates]
6        - !Sh |
7            cd /tmp
8            wget https://releases.hashicorp.com/consul/0.8.3/consul\_0.8.3\_
9            unzip consul_0.8.3_linux_amd64.zip
10           cp consul /usr/bin/consul
```

# vagga.yaml

```
1  containers:
2    ubuntu-consul:
3      setup:
4        - !Ubuntu xenial
5        - !Install [unzip, wget, ca-certificates]
6        - !Sh |
7            cd /tmp
8            wget https://releases.hashicorp.com/consul/0.8.3/consul\_0.8.3\_
9            unzip consul_0.8.3_linux_amd64.zip
10           cp consul /usr/bin/consul
11  commands:
12    consul-server: !Command
```

# vagga.yaml

```
1  containers:
2    ubuntu-consul:
3      setup:
4        - !Ubuntu xenial
5        - !Install [unzip, wget, ca-certificates]
6        - !Sh |
7            cd /tmp
8            wget https://releases.hashicorp.com/consul/0.8.3/consul\_0.8.3\_
9            unzip consul_0.8.3_linux_amd64.zip
10           cp consul /usr/bin/consul
11  commands:
12    consul-server: !Command
13      description: Start consul in server mode
14      container: ubuntu-consul
```

# vagga.yaml

```
1  containers:
2    ubuntu-consul:
3      setup:
4        - !Ubuntu xenial
5        - !Install [unzip, wget, ca-certificates]
6        - !Sh |
7            cd /tmp
8            wget https://releases.hashicorp.com/consul/0.8.3/consul\_0.8.3\_
9            unzip consul_0.8.3_linux_amd64.zip
10           cp consul /usr/bin/consul
11  commands:
12    consul-server: !Command
13      description: Start consul in server mode
14      container: ubuntu-consul
15      run: |
16          /usr/bin/consul agent -server -bootstrap-expect=1 \
17              -data-dir=/tmp/consul -log-level=debug \
18              -advertise=127.0.0.1
```



# *Dev-docker- compose*

# **.generate.yaml**



# .generate.yaml

```
1  version: '3'
2
3  services:
```



# .generate.yaml

```
1  version: '3'
```

```
2
```

```
3  services:
```

```
10  volumes:
```

```
11    consuldb:
```

```
12      external: true
```



# .generate.yaml

```
1  version: '3'
2
3  services:
4      # Key/Value storage with config for our app
5      consul: !include:consul.yaml |
6
7      # Search engine and document-based DB.
8      elasticsearch: !include:elasticsearch.yaml |
9
10 volumes:
11     consuldb:
12         external: true
```



# Dev-... vs docker-compose

```
1  version: '3'
2
3  services:
4    # Add Key/Value to Consul.
5    add_keys_to_consul: !include:add_keys_to_consul.yaml |
6
7    # Key/Value storage with config for our app and other infrastructure.
8    consul: !include:consul.yaml |
9
10   # Search engine and document-based DB.
11   elasticsearch: !include:elasticsearch.yaml |
12
13   # Add logs from filesystem to Elasticsearch.
14   filebeat: !include:filebeat.yaml |
15
16   # Visualization tool for data in Elasticsearch.
17   kibana: !include:kibana.yaml |
18
19   # Queue with logs which should write to DB.
20   rabbit: !include:rabbit.yaml |
21
22   # Consul services registrator.
23   registrator: !include:registrator.yaml |
24
25
26  volumes:
27    rabbitdb:
28      external: true
29    consuldb:
30      external: true
31
32  networks:
33    default:
34      external:
35        name: dev-compose-network
```

# Dev-... vs docker-compose

```
1  version: '3'
2
3  services:
4    # Add Key/Value to Consul.
5    add_keys_to_consul: !include:add_keys_to_consul.yaml |
6
7    # Key/Value storage with config for our app and other infrastructure.
8    consul: !include:consul.yaml |
9
10   # Search engine and document-based DB.
11   elasticsearch: !include:elasticsearch.yaml |
12
13   # Add logs from filesystem to Elasticsearch.
14   filebeat: !include:filebeat.yaml |
15
16   # Visualization tool for data in Elasticsearch.
17   kibana: !include:kibana.yaml |
18
19   # Queue with logs which should write to DB.
20   rabbit: !include:rabbit.yaml |
21
22   # Consul services registrator.
23   registrator: !include:registrator.yaml |
24
25
26  volumes:
27    rabbitdb:
28      external: true
29    consuldb:
30      external: true
31
32  networks:
33    default:
34      external:
35        name: dev-compose-network
```

```
1  networks:
2    default:
3      external:
4        name: dev-compose-network
5  services:
6    add_keys_to_consul:
7      build: ../dev_docker_compose/yaml_files/add_keys_to_consul
8      depends_on:
9        - consul
10     environment:
11       CONSUL_SERVICE_NAME: dev-app
12       REGION: us-east-1
13     network_mode: host
14     restart: on-failure
15     volumes:
16       - ../consul_conf:/consul_conf/
17   consul:
18     build: ../dev_docker_compose/yaml_files/consul
19     network_mode: host
20     restart: always
21     volumes:
22       - consuldb:/consul/data
23   elasticsearch:
24     deploy:
25       resources:
26         limits:
27           memory: 1g
28     environment:
29       - cluster.name=EScluster
30       - bootstrap.memory_lock=true
31       - ES_JAVA_OPTS=-Xms512m -Xmx512m
32       - xpack.security.enabled=false
33     image: docker.elastic.co/elasticsearch/elasticsearch:5.4.0
34     network_mode: host
35     ports:
36       - 9200:9200
37     restart: always
38     ulimits:
39       memlock:
40         hard: -1
41         soft: -1
42   filebeat:
43     build: ../dev_docker_compose/yaml_files/filebeat
44     depends_on:
45       - elasticsearch
46     environment:
47       CONSUL_SERVICE_NAME: dev-app
48     network_mode: host
49     restart: always
50     volumes:
51       - ../elastic_logs:/elastic_logs/
52   kibana:
53     environment:
54       ELASTICSEARCH_URL: http://localhost:9200
55       SERVER_NAME: http://localhost:5601
56       XPACK_MONITORING_UI_CONTAINER_ELASTICSEARCH_ENABLED: 'false'
57     image: docker.elastic.co/kibana/kibana:5.4.0
58     network_mode: host
59     restart: always
60   rabbit:
61     domainname: rabbit.local
62     hostname: rabbitmq
63     image: pdffiller/rabbitmq:latest
64     network_mode: host
65     ports:
66       - 4369:4369
67       - 5671:5671
68       - 5672:5672
69       - 15671:15671
70       - 15672:15672
71       - 25672:25672
72     restart: always
73     volumes:
74       - rabbitdb:/var/lib/rabbitmq
75   registrator:
76     command: consul://127.0.0.1:8500
77     depends_on:
78       - consul
79     image: gliderlabs/registrator:latest
80     network_mode: host
81     restart: always
82     volumes:
83       - /var/run/docker.sock:/tmp/docker.sock
84   version: '3'
85  volumes:
86    consuldb:
87      external: true
88    rabbitdb:
89      external: true
90
```

# Dev-... vs docker-compose

```
1  version: '3'
2
3  services:
4    # Add Key/Value to Consul.
5    add_keys_to_consul: !include:add_keys_to_consul.yaml |
6
7    # Key/Value storage with config for our app and other
8    consul: !include:consul.yaml |
9
10   # Search engine and document-based DB.
11   elasticsearch: !include:elasticsearch.yaml |
12
13   # Add logs from filesystem to Elasticsearch.
14   filebeat: !include:filebeat.yaml |
15
16   # Visualization tool for data in Elasticsearch.
17   kibana: !include:kibana.yaml |
18
19   # Queue with logs which should write to DB.
20   rabbit: !include:rabbit.yaml |
21
22   # Consul services registrator.
23   registrator: !include:registrator.yaml |
24
25
26  volumes:
27    rabbitdb:
28      external: true
29    consuldb:
30      external: true
31
32  networks:
33    default:
34      external:
35        name: dev-compose-network
```

```
1  version: '3'
2
3  services:
4    # Add Key/Value to Consul.
5    add_keys_to_consul: !include:add_keys_to_consul.yaml |
6
7    # Key/Value storage with config for our app and other infrastructure.
8    consul: !include:consul.yaml |
9
10   # Search engine and document-based DB.
11   elasticsearch: !include:elasticsearch.yaml |
12
13   # Add logs from filesystem to Elasticsearch.
14   filebeat: !include:filebeat.yaml |
15
16   # Visualization tool for data in Elasticsearch.
17   kibana: !include:kibana.yaml |
18
19   # Queue with logs which should write to DB.
20   rabbit: !include:rabbit.yaml |
21
22   # Consul services registrator.
23   registrator: !include:registrator.yaml |
24
25
26  volumes:
27    rabbitdb:
28      external: true
29    consuldb:
30      external: true
31
32  networks:
33    default:
34      external:
35        name: dev-compose-network
```

```
1  networks:
2    default:
3      external:
4        name: dev-compose-network
5
6  services:
7    add_keys_to_consul:
8      build: ../dev_docker_compose/yaml_files/add_keys_to_consul
9      depends_on:
10        - consul
11      environment:
12        CONSUL_SERVICE_NAME: dev-app
13        REGION: us-east-1
14      network_mode: host
15      restart: on-failure
16      volumes:
17        - ../consul_conf:/consul_conf/
18    consul:
19      build: ../dev_docker_compose/yaml_files/consul
20      network_mode: host
21      restart: always
22      volumes:
23        - consuldb:/consul/data
24    elasticsearch:
25      deploy:
26        resources:
27          limits:
28            memory: 1g
29      environment:
30        - cluster.name=EScluster
31        - bootstrap.memory_lock=true
32        - ES_JAVA_OPTS=-Xms512m -Xmx512m
33        - xpack.security.enabled=false
34        - node.name=es_node1
35      image: docker.elastic.co/elasticsearch/elasticsearch:5.4.0
36      network_mode: host
37      ports:
38        - 9200:9200
39      restart: always
40      ulimits:
41        memlock:
42          hard: -1
43          soft: -1
44    filebeat:
45      build: ../dev_docker_compose/yaml_files/filebeat
46      depends_on:
47        - elasticsearch
48      environment:
49        CONSUL_SERVICE_NAME: dev-app
50      network_mode: host
51      restart: always
52      volumes:
53        - ../elastic_logs:/elastic_logs/
54    kibana:
55      environment:
56        ELASTICSEARCH_URL: http://localhost:9200
57        SERVER_NAME: http://localhost:5601
58        XPACK_MONITORING_UI_CONTAINER_ELASTICSEARCH_ENABLED: 'false'
59      image: docker.elastic.co/kibana/kibana:5.4.0
60      network_mode: host
61      restart: always
62    rabbit:
63      domainname: rabbit.local
64      hostname: rabbitmq
65      image: pdffiller/rabbitmq:latest
66      network_mode: host
67      ports:
68        - 4369:4369
69        - 5671:5671
70        - 5672:5672
71        - 15671:15671
72        - 15672:15672
73        - 25672:25672
74      restart: always
75      volumes:
76        - rabbitdb:/var/lib/rabbitmq
77    registrator:
78      command: consul://127.0.0.1:8500
79      depends_on:
80        - consul
81      image: gliderlabs/registrator:latest
82      network_mode: host
83      restart: always
84      volumes:
85        - /var/run/docker.sock:/tmp/docker.sock
86  version: '3'
87
88  volumes:
89    consuldb:
90      external: true
91    rabbitdb:
92      external: true
```

# Structure

- 1 dev-docker-compose
- 2 |— .generate.yaml



# Structure

```
1 dev-docker-compose
2 |— .generate.yaml
3 |— Makefile
4 |— Makefile.dev.mk
```





# Structure

```
1 dev-docker-compose
2 |— .generate.yaml
3 |— Makefile
4 |— Makefile.dev.mk
5 |— .dev_docker_compose
6 |   |— Dockerfile
7 |   |— loader.py
8 |   |— RELEASE_NOTES.md
```



# Structure

```
1  dev-docker-compose
2  |— .generate.yaml
3  |— Makefile
4  |— Makefile.dev.mk
5  |— .dev_docker_compose
6  |   |— Dockerfile
7  |   |— loader.py
8  |   |— RELEASE_NOTES.md
9  |   └─ yaml_files
10 |       |— consul.yaml
11 |       └─ elasticsearch.yaml
```

# Structure

```
1  dev-docker-compose
2  |— .generate.yaml
3  |— Makefile
4  |— Makefile.dev.mk
5  |— .dev_docker_compose
6  |   |— Dockerfile
7  |   |— loader.py
8  |   |— RELEASE_NOTES.md
9  |   |— yaml_files
10 |       |— consul.yaml
11 |       |— elasticsearch.yaml
12 |— .consul_conf
13 |   |— dev_conf.json
14 |   |— devops_conf.json
```

# Structure

```
1 dev-docker-compose
2 |— .generate.yaml
3 |— Makefile
4 |— Makefile.dev.mk
5 |— .dev_docker_compose
6 |   |— Dockerfile
7 |   |— loader.py
8 |   |— RELEASE_NOTES.md
9 |   |— yaml_files
10 |       |— consul.yaml
11 |       |— elasticsearch.yaml
12 |— .consul_conf
13 |   |— dev_conf.json
14 |   |— devops_conf.json
15 |— .elastic_logs
16 |   |— logs.json
```

# Structure

```
1  dev-docker-compose
2  |— .generate.yaml
3  |— Makefile
4  |— Makefile.dev.mk
5  |— .dev_docker_compose
6  |   |— Dockerfile
7  |   |— loader.py
8  |   |— RELEASE_NOTES.md
9  |   └─ yaml_files
10 |       |— consul.yaml
11 |       └─ elasticsearch.yaml
12 |— .consul_conf
13 |   |— dev_conf.json
14 |   └─ devops_conf.json
15 |— .elastic_logs
16 |   └─ logs.json
17 └─ README.md
```

# *Questions?*

*fb.com/maxymvlasov*  
*@maxymvlasov*



*bit.ly/msp-docker*