

Web-based behavioral experiments for online data collection

Paxton Fitzpatrick
November 6, 2020

Overview

Overview

Goal: Create a behavioral experiment that:

Overview

Goal: Create a behavioral experiment that:

- can be run both in-lab and online

Overview

Goal: Create a behavioral experiment that:

- can be run both in-lab and online
- handles complex stimuli and responses

Overview

Goal: Create a behavioral experiment that:

- can be run both in-lab and online
- handles complex stimuli and responses
- is self-isolated and efficient

Overview

Goal: Create a behavioral experiment that:

- can be run both in-lab and online
- handles complex stimuli and responses
- is self-isolated and efficient
- works on any computer

Overview

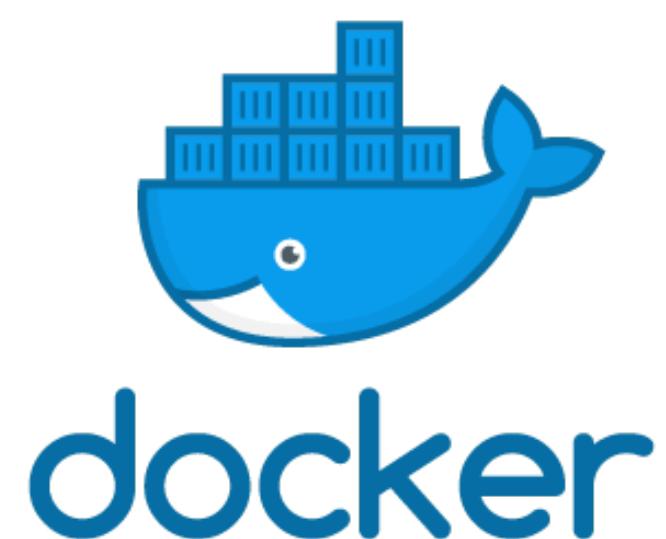
Goal: Create a behavioral experiment that:

- can be run both in-lab and online
- handles complex stimuli and responses
- is self-isolated and efficient
- works on any computer
- is easy to design and test

Overview

Goal: Create a behavioral experiment that:

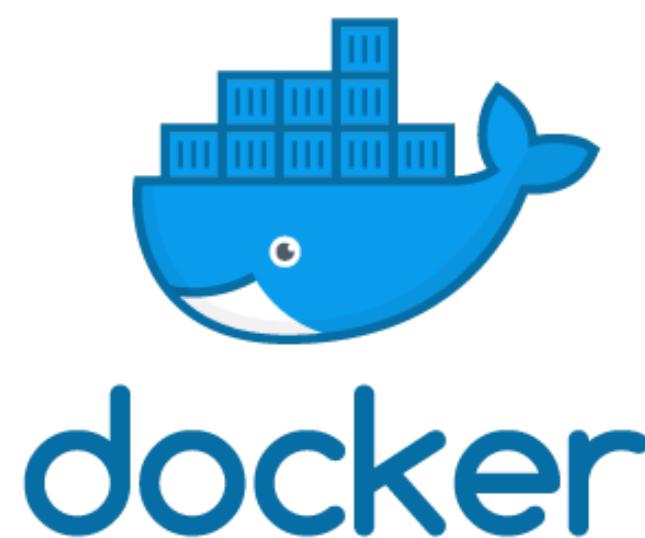
- can be run both in-lab and online
- handles complex stimuli and responses
- is self-isolated and efficient
- works on any computer
- is easy to design and test



Overview

Goal: Create a behavioral experiment that:

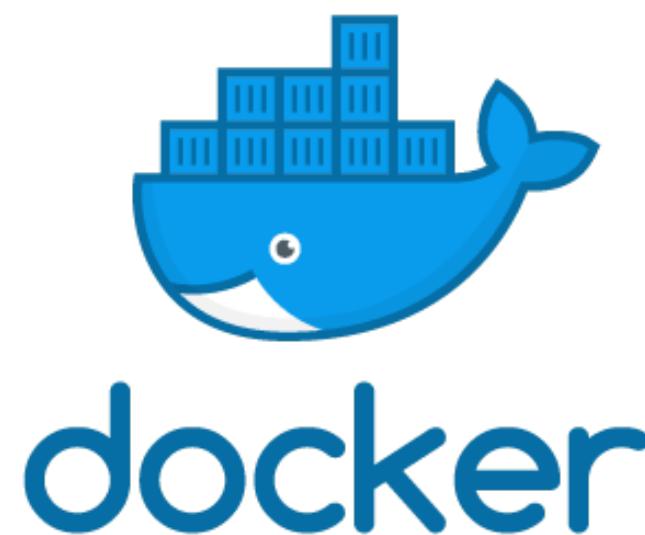
- can be run both in-lab and online
- handles complex stimuli and responses
- is self-isolated and efficient
- works on any computer
- is easy to design and test



Overview

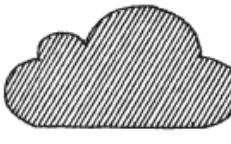
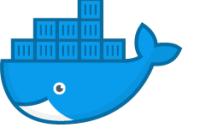
Goal: Create a behavioral experiment that:

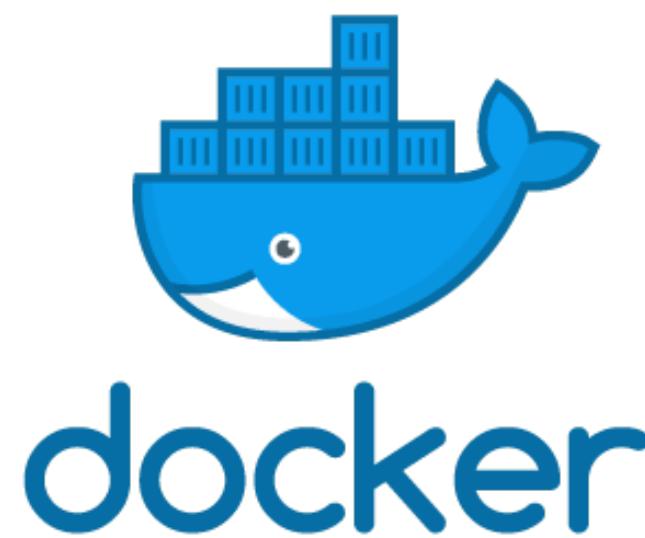
- can be run both in-lab and online
- handles complex stimuli and responses
- is self-isolated and efficient
- works on any computer
- is easy to design and test



Overview

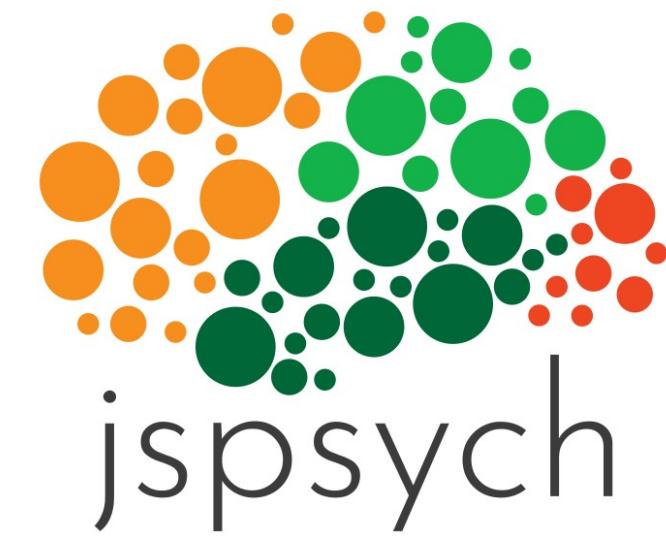
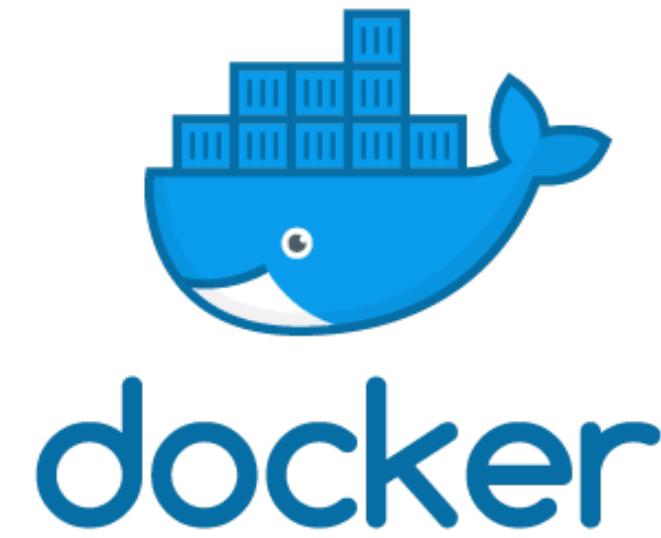
Goal: Create a behavioral experiment that:

- can be run both in-lab and online 
- handles complex stimuli and responses 
- is self-isolated and efficient 
- works on any computer 
- is easy to design and test 



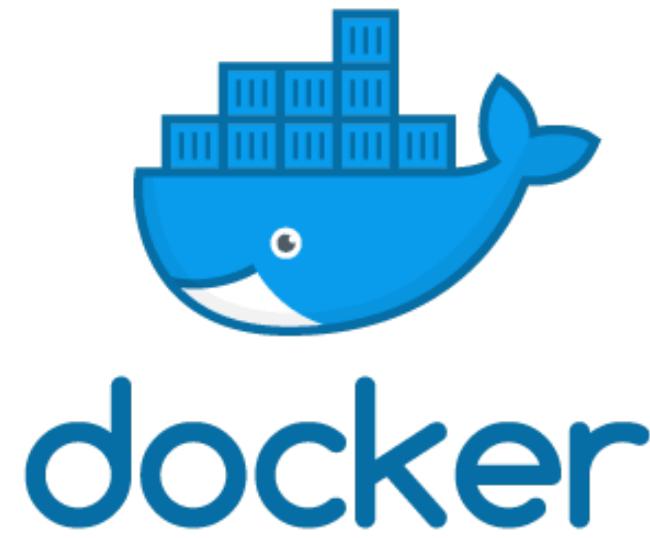
Overview

- Brief overview of framework: 3 powerful tools



Overview

- Brief overview of framework: 3 powerful tools

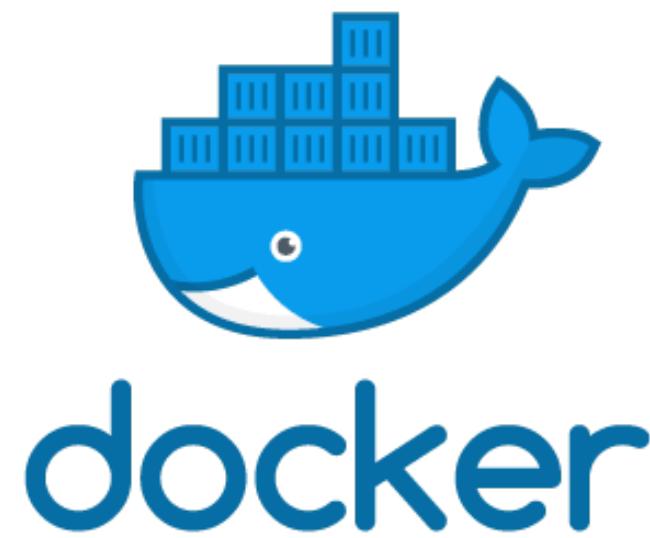


- Create an experiment ready for online deployment



Overview

- Brief overview of framework: 3 powerful tools



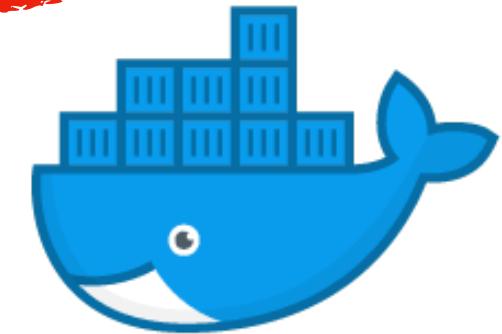
- Create an experiment ready for online deployment



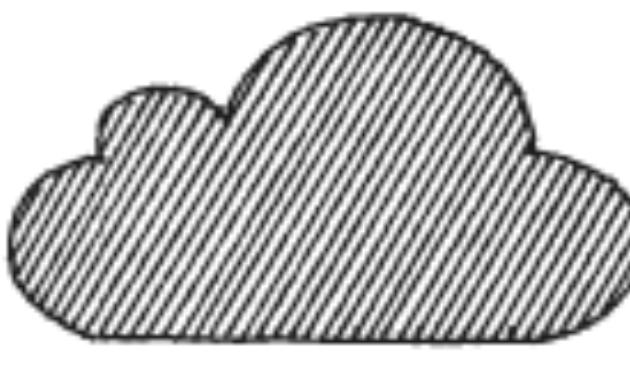
- Live demo (👉)

Overview

- Brief overview of framework: 3 powerful tools



docker



PSITURK



- Create an experiment ready for online deployment

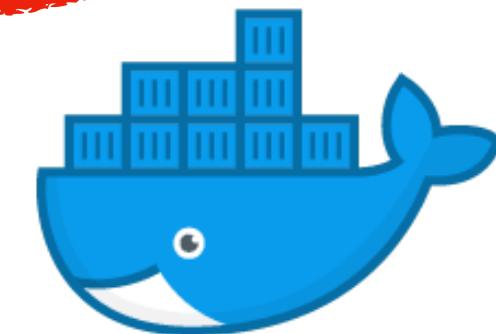
amazon

mechanical turk

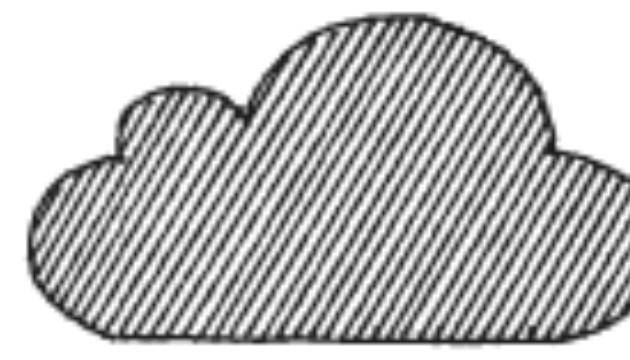
- Live demo (👉)

Overview

- Brief overview of framework: 3 powerful tools



docker



PSITURK

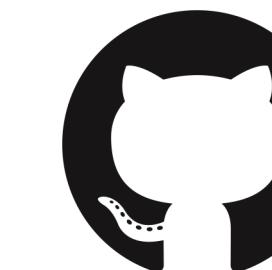


- Create an experiment ready for online deployment

amazon

mechanical turk

- Live demo (👉)



github.com/paxtonfitzpatrick/psiturk-experiment-template



What is MTurk?



What is MTurk?

- Crowdsourcing platform created by Amazon



What is MTurk?

- Crowdsourcing platform created by Amazon
- “*Requesters*” post “*HITs*” (Human Intelligence Tasks) for “*workers*” to complete



What is MTurk?

- Crowdsourcing platform created by Amazon
- “*Requesters*” post “*HITs*” (Human Intelligence Tasks) for “*workers*” to complete
- HITs range from simple to complex



What is MTurk?

- Crowdsourcing platform created by Amazon
- “*Requesters*” post “*HITs*” (Human Intelligence Tasks) for “*workers*” to complete
- HITs range from simple to complex
 - Image labeling for ML



What is MTurk?

- Crowdsourcing platform created by Amazon
- “*Requesters*” post “*HITs*” (Human Intelligence Tasks) for “*workers*” to complete
- HITs range from simple to complex
 - Image labeling for ML
 - Audio transcription



What is MTurk?

- Crowdsourcing platform created by Amazon
- “*Requesters*” post “*HITs*” (Human Intelligence Tasks) for “*workers*” to complete
- HITs range from simple to complex
 - Image labeling for ML
 - Audio transcription
 - Writing product descriptions for websites



What is MTurk?

- Crowdsourcing platform created by Amazon
- “*Requesters*” post “*HITs*” (Human Intelligence Tasks) for “*workers*” to complete
- HITs range from simple to complex
 - Image labeling for ML
 - Audio transcription
 - Writing product descriptions for websites
 - Full behavioral experiments



Why use MTurk?



Why use MTurk?

- Large participant pool...



Why use MTurk?

- **Large participant pool...**
 - 100,000 – 200,000 MTurk workers¹

Why use MTurk?

- **Large participant pool...**
 - 100,000 – 200,000 MTurk workers¹
 - 2,000–12,000 active online at any time¹

Why use MTurk?

- **Large participant pool...**
 - 100,000 – 200,000 MTurk workers¹
 - 2,000–12,000 active online at any time¹
 - Typical lab can reach to 7,300 participants per posting²

¹Difallah et al., 2018

²Stewart et al., 2015

Why use MTurk?

- **Large participant pool...**
 - 100,000 – 200,000 MTurk workers¹
 - 2,000–12,000 active online at any time¹
 - Typical lab can reach to 7,300 participants per posting²
 - 50% of pool refreshed every 7 months²

¹Difallah et al., 2018

²Stewart et al., 2015

Why use MTurk?

- **Large participant pool...**
 - 100,000 – 200,000 MTurk workers¹
 - 2,000–12,000 active online at any time¹
 - Typical lab can reach to 7,300 participants per posting²
 - 50% of pool refreshed every 7 months²
- **...enables fast data collection**

¹Difallah et al., 2018

²Stewart et al., 2015

Why use MTurk?

- **Large participant pool...**
 - 100,000 – 200,000 MTurk workers¹
 - 2,000–12,000 active online at any time¹
 - Typical lab can reach to 7,300 participants per posting²
 - 50% of pool refreshed every 7 months²
- **...enables fast data collection**
 - 30,002 participants in 6 weeks (no restrictions)³

Why use MTurk?

- **Large participant pool...**
 - 100,000 – 200,000 MTurk workers¹
 - 2,000–12,000 active online at any time¹
 - Typical lab can reach to 7,300 participants per posting²
 - 50% of pool refreshed every 7 months²
- **...enables fast data collection**
 - 30,002 participants in 6 weeks (no restrictions)³
 - 9,770 participants in 8 weeks (US-only, “master workers”-only)⁴

Why use MTurk?

- **Large participant pool...**
 - 100,000 – 200,000 MTurk workers¹
 - 2,000–12,000 active online at any time¹
 - Typical lab can reach to 7,300 participants per posting²
 - 50% of pool refreshed every 7 months²
- **...enables fast data collection**
 - 30,002 participants in 6 weeks (no restrictions)³
 - 9,770 participants in 8 weeks (US-only, “master workers”-only)⁴
 - Personal experience: 10 participants in ~2 hours (US-only, “master workers”-only)



Why use MTurk?



Why use MTurk?

- Participant pool is diverse...

Why use MTurk?

- **Participant pool is diverse...**
 - Significantly more diverse than typical college samples^{5,6,7}

Why use MTurk?

- **Participant pool is diverse...**

- Significantly more diverse than typical college samples^{5,6,7}
- More diverse than community samples recruited from college towns⁸

Why use MTurk?

- **Participant pool is diverse...**

- Significantly more diverse than typical college samples^{5,6,7}
- More diverse than community samples recruited from college towns⁸
- Slightly more diverse than standard internet samples⁷

Why use MTurk?

- **Participant pool is diverse...**

- Significantly more diverse than typical college samples^{5,6,7}
- More diverse than community samples recruited from college towns⁸
- Slightly more diverse than standard internet samples⁷
- More representative of US population than in-person convenience samples⁹

Why use MTurk?

- **Participant pool is diverse...**
 - Significantly more diverse than typical college samples^{5,6,7}
 - More diverse than community samples recruited from college towns⁸
 - Slightly more diverse than standard internet samples⁷
 - More representative of US population than in-person convenience samples⁹
- **...and specific populations can be subsampled**

Why use MTurk?

- **Participant pool is diverse...**
 - Significantly more diverse than typical college samples^{5,6,7}
 - More diverse than community samples recruited from college towns⁸
 - Slightly more diverse than standard internet samples⁷
 - More representative of US population than in-person convenience samples⁹
- **...and specific populations can be subsampled**
 - Geographic area (US-only, Ohio-only, etc.)

Why use MTurk?

- **Participant pool is diverse...**
 - Significantly more diverse than typical college samples^{5,6,7}
 - More diverse than community samples recruited from college towns⁸
 - Slightly more diverse than standard internet samples⁷
 - More representative of US population than in-person convenience samples⁹
- **...and specific populations can be subsampled**
 - Geographic area (US-only, Ohio-only, etc.)
 - Other voluntarily provided info (age, ethnicity, marital status, handedness, online usage, etc.)



Why use MTurk?

Why use MTurk?

But what about data quality and validity?

Why use MTurk?

But what about data quality and validity?

- **Classic behavioral findings replicate well**

Why use MTurk?

But what about data quality and validity?

- **Classic behavioral findings replicate well**
 - Cognitive biases, logical fallacies, and strategic behavior¹⁰

Why use MTurk?

But what about data quality and validity?

- **Classic behavioral findings replicate well**
 - Cognitive biases, logical fallacies, and strategic behavior¹⁰
 - Response time-dependent measures (Stroop, switching, flanker, attentional blink, subliminal priming)¹¹

Why use MTurk?

But what about data quality and validity?

- **Classic behavioral findings replicate well**
 - Cognitive biases, logical fallacies, and strategic behavior¹⁰
 - Response time-dependent measures (Stroop, switching, flanker, attentional blink, subliminal priming)¹¹
 - MTurk data matches or exceeds in-person psychometric validity^{7,12}

Why use MTurk?

Why use MTurk?

But what about data quality and validity?

- **MTurk workers appear to be honest and engaged**

Why use MTurk?

But what about data quality and validity?

- **MTurk workers appear to be honest and engaged**
 - High test-retest reliability, comparable to in-person

Why use MTurk?

But what about data quality and validity?

- **MTurk workers appear to be honest and engaged**
 - High test-retest reliability, comparable to in-person
 - After 1 week¹²

Why use MTurk?

But what about data quality and validity?

- **MTurk workers appear to be honest and engaged**
 - High test-retest reliability, comparable to in-person
 - After 1 week¹²
 - After 3 weeks^{7,13}

Why use MTurk?

But what about data quality and validity?

- **MTurk workers appear to be honest and engaged**
 - High test-retest reliability, comparable to in-person
 - After 1 week¹²
 - After 3 weeks^{7,13}
 - After 6 months¹⁴

Why use MTurk?

But what about data quality and validity?

- **MTurk workers appear to be honest and engaged**
 - High test-retest reliability, comparable to in-person
 - After 1 week¹²
 - After 3 weeks^{7,13}
 - After 6 months¹⁴
 - Direct assessments of attentiveness are comparable to in-person^{5,15}

Why use MTurk?

But what about data quality and validity?

- **MTurk workers appear to be honest and engaged**
 - High test-retest reliability, comparable to in-person
 - After 1 week¹²
 - After 3 weeks^{7,13}
 - After 6 months¹⁴
 - Direct assessments of attentiveness are comparable to in-person^{5,15}
 - No more likely to cheat than college student samples¹⁶

Experiment structure

Experiment structure

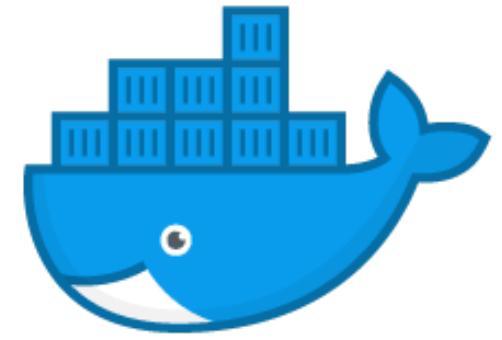
my_experiment/

Experiment structure

```
my_experiment/  
└── exp/
```

Experiment structure

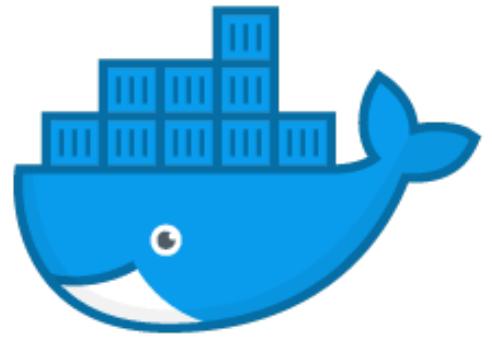
```
my_experiment/  
└── exp/  
└── data/
```



docker

What is Docker?

my_experiment/
└── exp/
 └── data/

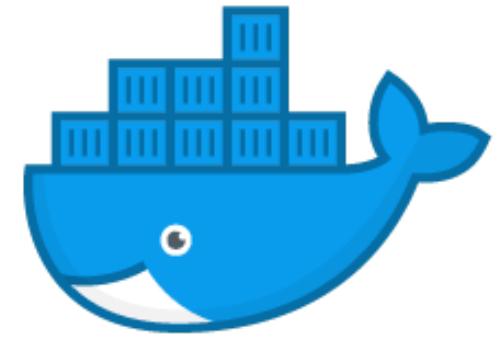


docker

What is Docker?

my_experiment/
└── exp/
 └── data/

- Self-contained, isolated software environments

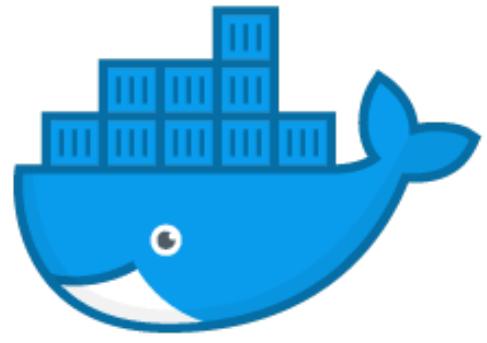


docker

What is Docker?

my_experiment/
└── exp/
 └── data/

- Self-contained, isolated software environments
- Create, share, and deploy applications & services

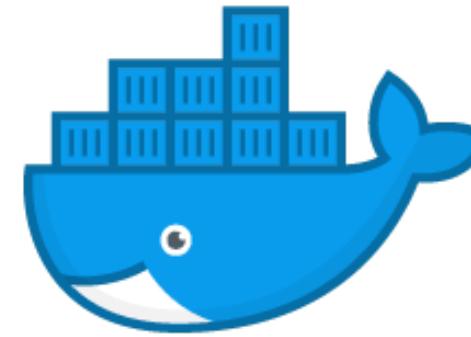


docker

What is Docker?

my_experiment/
└── exp/
 └── data/

- Self-contained, isolated software environments
- Create, share, and deploy applications & services
- Popular tool in production environments



docker

What is Docker?

my_experiment/
└── exp/
 └── data/

- Self-contained, isolated software environments
- Create, share, and deploy applications & services
- Popular tool in production environments

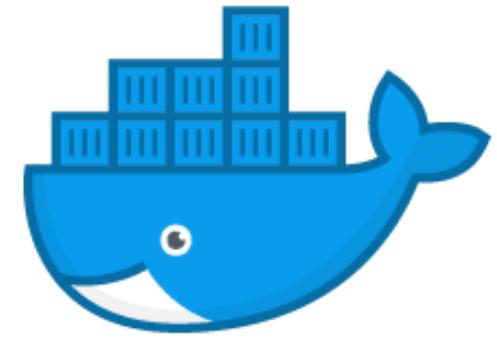


ebay



verizon^v

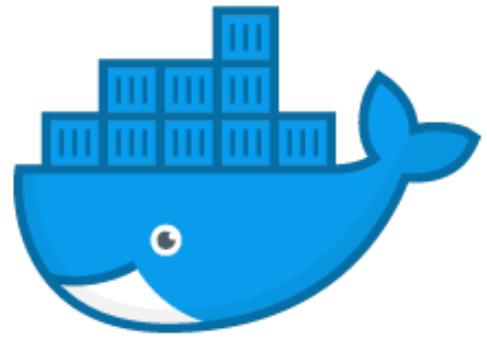
BBC



docker

Why use Docker?

my_experiment/
└── exp/
 └── data/

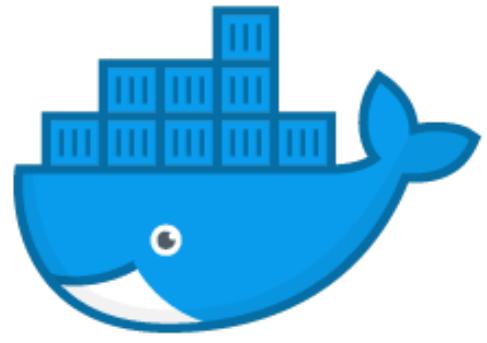


docker

Why use Docker?

my_experiment/
└── exp/
 └── data/

- Run multiple experiments on the same computer

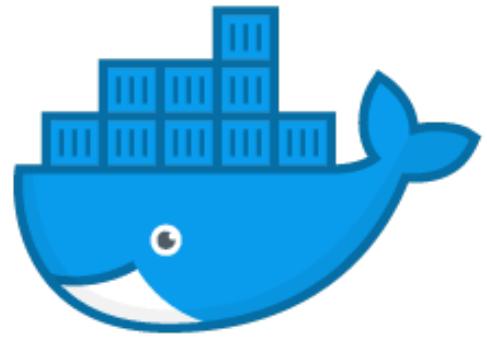


docker

my_experiment/
└── exp/
 └── data/

Why use Docker?

- Run multiple experiments on the same computer
- Avoid package, path, port, etc. conflicts

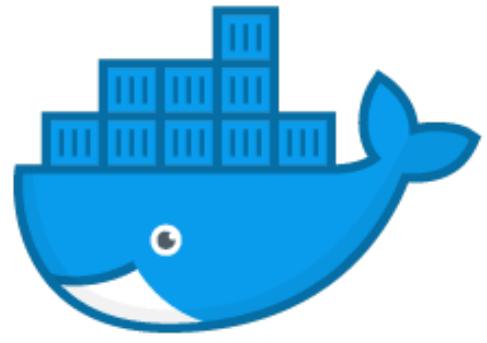


docker

my_experiment/
└── exp/
 └── data/

Why use Docker?

- Run multiple experiments on the same computer
- Avoid package, path, port, etc. conflicts
- Experiment will run the same on any computer

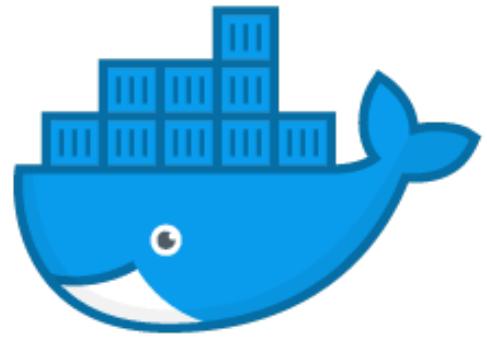


docker

my_experiment/
└── exp/
 └── data/

Why use Docker?

- Run multiple experiments on the same computer
- Avoid package, path, port, etc. conflicts
- Experiment will run the same on any computer
- Great for sharing reproducible experiment code

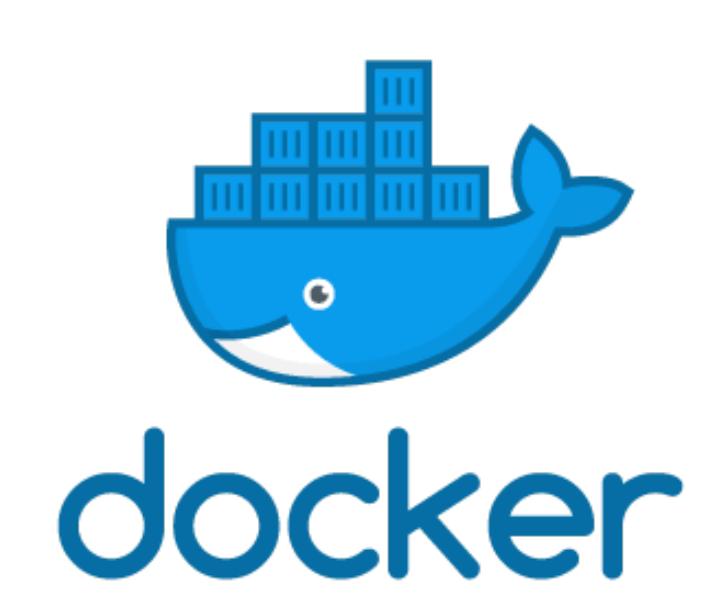


docker

my_experiment/
└── exp/
 └── data/

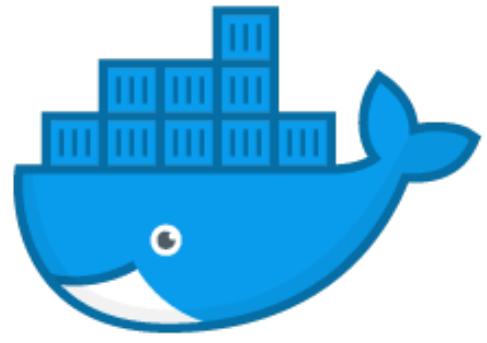
Why use Docker?

- Run multiple experiments on the same computer
- Avoid package, path, port, etc. conflicts
- Experiment will run the same on any computer
- Great for sharing reproducible experiment code
- Easily manage all experiment-related services



Core concepts

```
my_experiment/  
└── exp/  
    └── data/
```



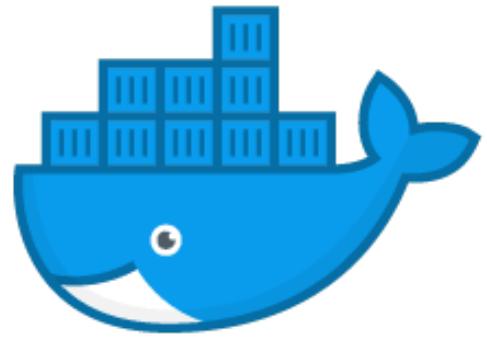
docker

my_experiment/
└── exp/
 └── data/

Core concepts

Image

“Template” environment with everything needed to perform a certain task



docker

Core concepts

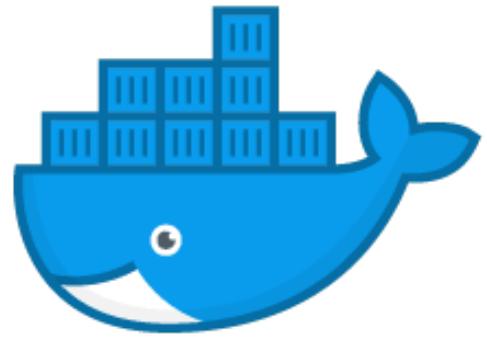
my_experiment/
└── exp/
 └── data/

Image

“Template” environment with everything needed to perform a certain task

Container

A running instance of an image



docker

my_experiment/
└── exp/
 └── data/

Core concepts

Image

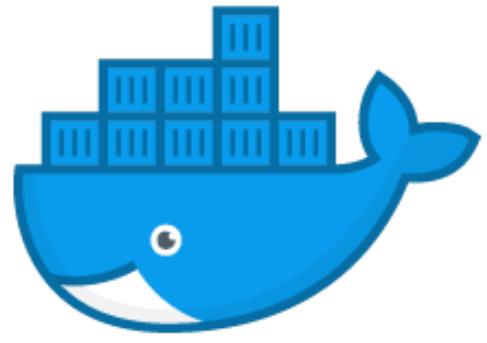
“Template” environment with everything needed to perform a certain task

Container

A running instance of an image

Dockerfile

File containing instructions to build an image



docker

Core concepts

my_experiment/
└── exp/
 └── data/

Image

“Template” environment with everything needed to perform a certain task

Container

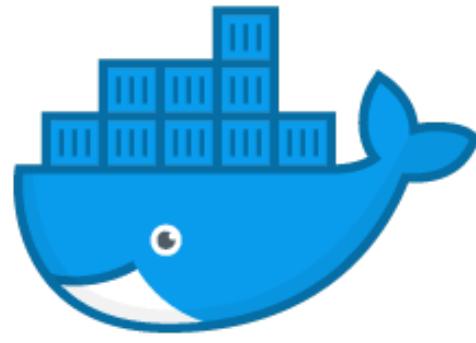
A running instance of an image

Dockerfile

File containing instructions to build an image

docker-compose

Tool for defining and running multiple, coordinated containers



docker

Core concepts

my_experiment/
└── exp/
 └── data/

Image

“Template” environment with everything needed to perform a certain task

Container

A running instance of an image

Dockerfile

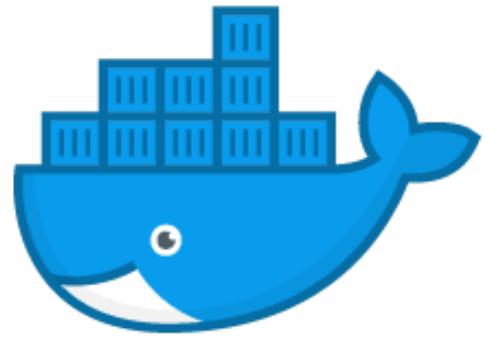
File containing instructions to build an image

docker-compose

Tool for defining and running multiple, coordinated containers

Docker Hub

GitHub-like site with repositories of pre-built images

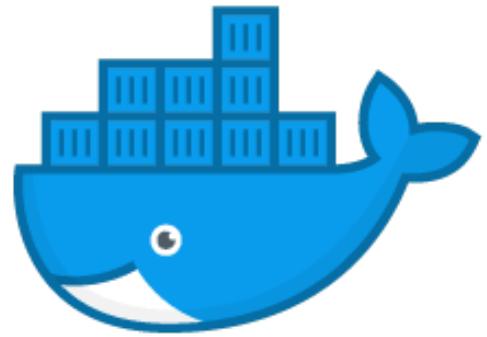


docker

Dockerfile

my_experiment/
└── exp/
 └── data/

```
● ● ●  
FROM python:3.6-stretch  
ENV PSITURK_GLOBAL_CONFIG_LOCATION "/exp"  
WORKDIR "/exp"  
RUN pip install \  
      psiturk==2.3.8 \  
      pymysql==0.10.0 \  
      python-Levenshtein==0.12.0 \  
      && rm -rf ~/.cache/pip  
CMD ["bash"]
```



docker

Dockerfile

my_experiment/
└── exp/
 └── data/



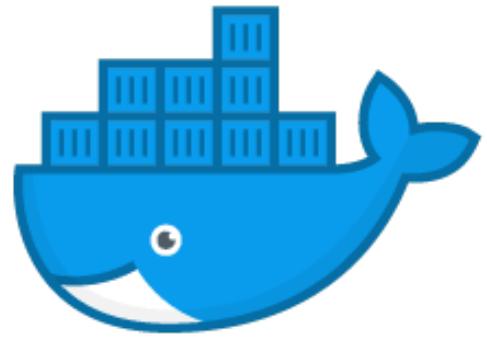
```
FROM python:3.6-stretch

ENV PSITURK_GLOBAL_CONFIG_LOCATION "/exp"

WORKDIR "/exp"

RUN pip install \
    psiturk==2.3.8 \
    pymysql==0.10.0 \
    python-Levenshtein==0.12.0 \
    && rm -rf ~/.cache/pip

CMD [ "bash" ]
```



docker

Dockerfile

my_experiment/
└── exp/
 └── data/

- Specify the base image to build from



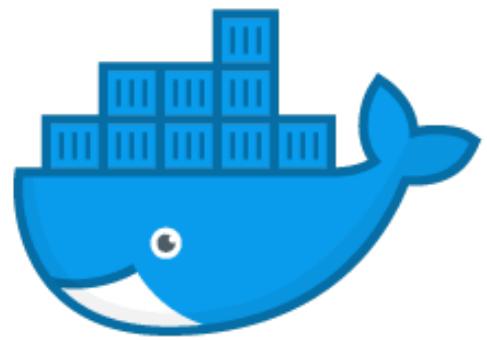
```
FROM python:3.6-stretch

ENV PSITURK_GLOBAL_CONFIG_LOCATION "/exp"

WORKDIR "/exp"

RUN pip install \
    psiturk==2.3.8 \
    pymysql==0.10.0 \
    python-Levenshtein==0.12.0 \
    && rm -rf ~/.cache/pip

CMD [ "bash" ]
```

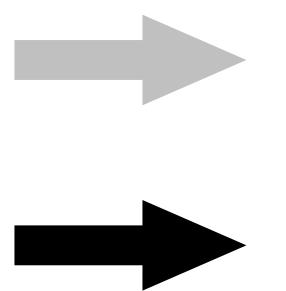


docker

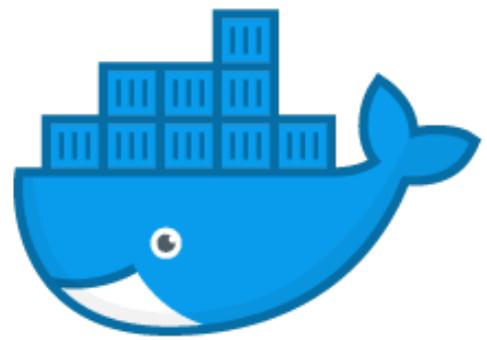
Dockerfile

my_experiment/
└── exp/
 └── data/

- Specify the base image to build from
- Set environment variable (tells psiTurk where to look for `.psiturkconfig`)



```
FROM python:3.6-stretch
ENV PSITURK_GLOBAL_CONFIG_LOCATION "/exp"
WORKDIR "/exp"
RUN pip install \
    psiturk==2.3.8 \
    pymysql==0.10.0 \
    python-Levenshtein==0.12.0 \
    && rm -rf ~/.cache/pip
CMD [ "bash" ]
```

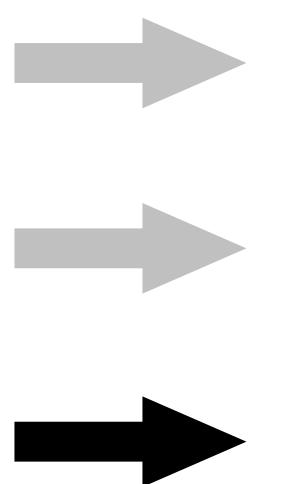


docker

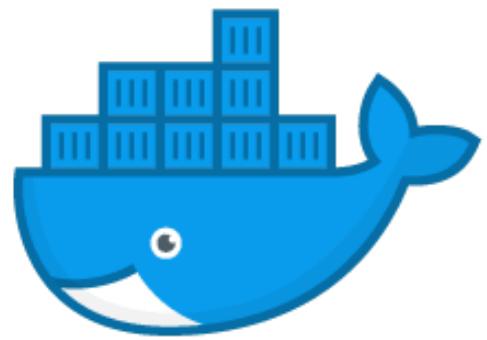
Dockerfile

my_experiment/
└── exp/
 └── data/

- Specify the base image to build from
- Set environment variable (tells psiTurk where to look for `.psiturkconfig`)
- Set the working directory for running containers



```
FROM python:3.6-stretch  
  
ENV PSITURK_GLOBAL_CONFIG_LOCATION "/exp"  
  
WORKDIR "/exp"  
  
RUN pip install \  
    psiturk==2.3.8 \  
    pymysql==0.10.0 \  
    python-Levenshtein==0.12.0 \  
    && rm -rf ~/.cache/pip  
  
CMD [ "bash" ]
```

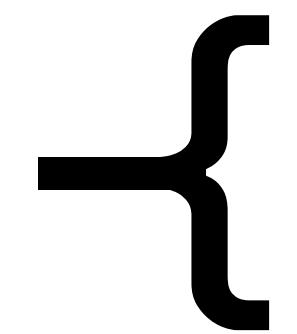
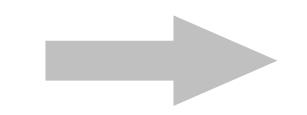
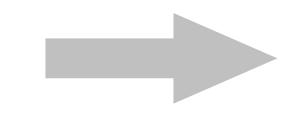
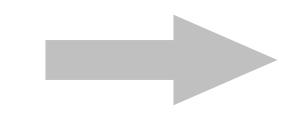


docker

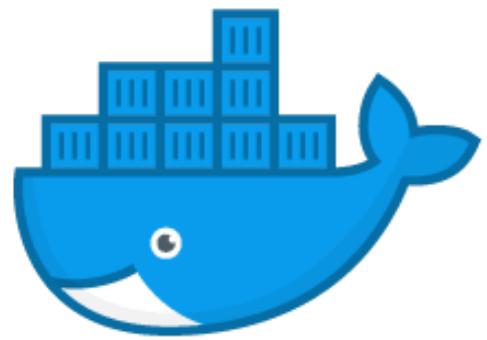
Dockerfile

my_experiment/
└── exp/
 └── data/

- Specify the base image to build from
- Set environment variable (tells psiTurk where to look for `.psiturkconfig`)
- Set the working directory for running containers
- Install psiTurk and some extra required packages



```
FROM python:3.6-stretch  
  
ENV PSITURK_GLOBAL_CONFIG_LOCATION "/exp"  
  
WORKDIR "/exp"  
  
RUN pip install \  
    psiturk==2.3.8 \  
    pymysql==0.10.0 \  
    python-Levenshtein==0.12.0 \  
    && rm -rf ~/.cache/pip  
  
CMD [ "bash" ]
```



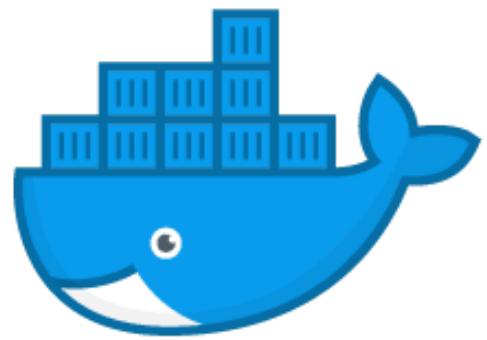
docker

Dockerfile

```
my_experiment/  
└── exp/  
    └── data/
```

- Specify the base image to build from
- Set environment variable (tells psiTurk where to look for `.psiturkconfig`)
- Set the working directory for running containers
- Install psiTurk and some extra required packages
- Clearing caches helps keep image small

```
FROM python:3.6-stretch  
  
ENV PSITURK_GLOBAL_CONFIG_LOCATION "/exp"  
  
WORKDIR "/exp"  
  
RUN pip install \  
    psiturk==2.3.8 \  
    pymysql==0.10.0 \  
    python-Levenshtein==0.12.0 \  
    && rm -rf ~/.cache/pip  
  
CMD [ "bash" ]
```



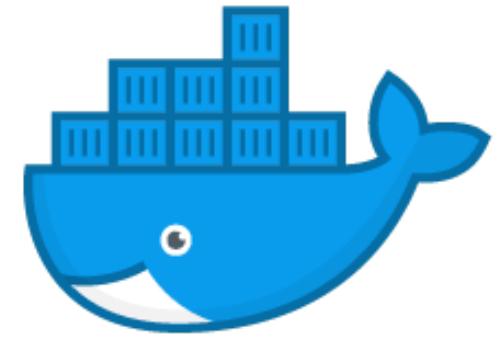
docker

Dockerfile

my_experiment/
└── exp/
 └── data/

- Specify the base image to build from
- Set environment variable (tells psiTurk where to look for `.psiturkconfig`)
- Set the working directory for running containers
- Install psiTurk and some extra required packages
- Clearing caches helps keep image small
- Set the command executed when running a container (launch a bash shell)

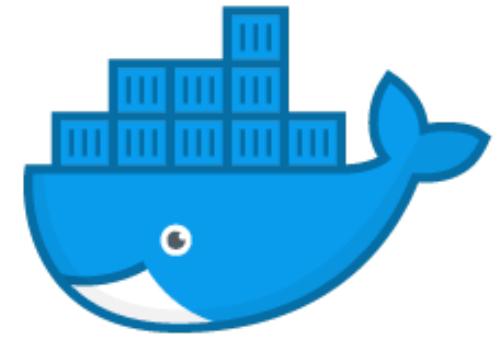
```
FROM python:3.6-stretch
ENV PSITURK_GLOBAL_CONFIG_LOCATION "/exp"
WORKDIR "/exp"
RUN pip install \
    psiturk==2.3.8 \
    pymysql==0.10.0 \
    python-Levenshtein==0.12.0 \
    && rm -rf ~/.cache/pip
CMD ["bash"]
```



docker

docker-compose.yml

my_experiment/
└── exp/
 └── data/

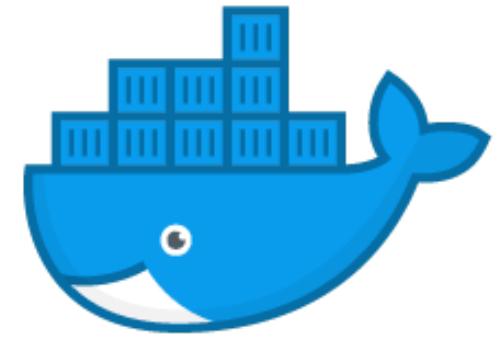


docker

docker-compose.yml

my_experiment/
└── exp/
 └── data/

- We need 3 support services to run on MTurk

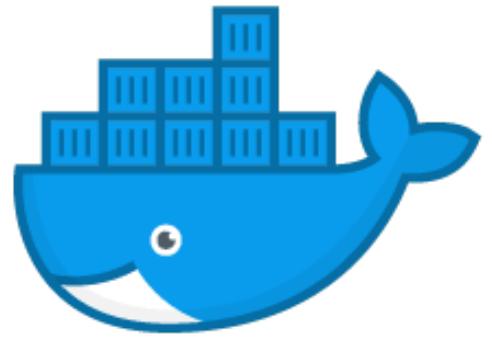


docker

docker-compose.yml

my_experiment/
└── exp/
 └── data/

- We need 3 support services to run on MTurk
- Normally complex to configure, tedious to manage

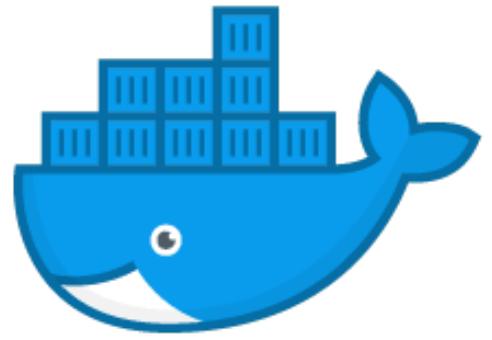


docker

docker-compose.yml

my_experiment/
└── exp/
 └── data/

- We need 3 support services to run on MTurk
- Normally complex to configure, tedious to manage
- But Docker makes this *much* easier!

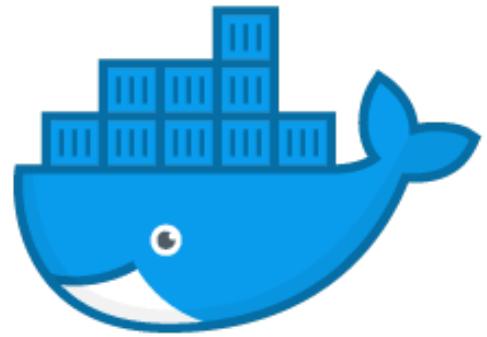


docker

docker-compose.yml

my_experiment/
└── exp/
 └── data/

- We need 3 support services to run on MTurk
- Normally complex to configure, tedious to manage
- But Docker makes this *much* easier!
 - All 3 available on Docker Hub as ready-to-run images

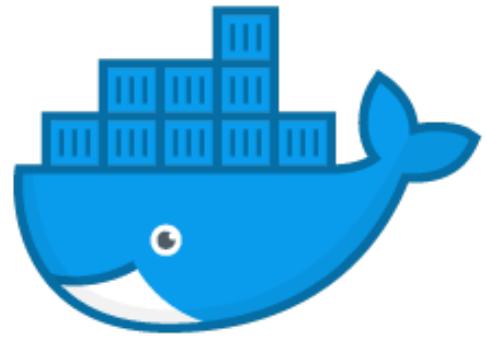


docker

docker-compose.yml

my_experiment/
└── exp/
 └── data/

- We need 3 support services to run on MTurk
- Normally complex to configure, tedious to manage
- But Docker makes this *much easier!*
 - All 3 available on Docker Hub as ready-to-run images
 - docker-compose: build, start and stop entire setup with a single command

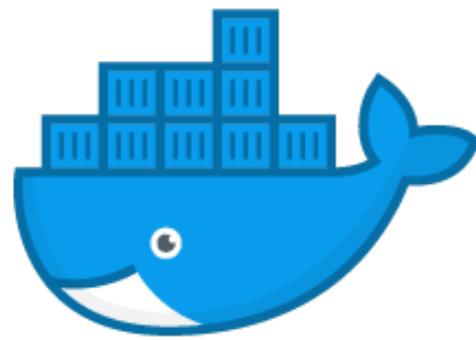


docker

docker-compose.yml

my_experiment/
└── exp/
 └── data/

- We need 3 support services to run on MTurk
- Normally complex to configure, tedious to manage
- But Docker makes this *much easier!*
 - All 3 available on Docker Hub as ready-to-run images
 - docker-compose: build, start and stop entire setup with a single command
 - Configure full application in docker-compose.yml



docker

docker-compose.yml

my_experiment/
└── exp/
 └── data/

- We need 3 support services to run on MTurk
- Normally complex to configure, tedious to manage
- But Docker makes this *much easier!*
 - All 3 available on Docker Hub as ready-to-run images
 - docker-compose: build, start and stop entire setup with a single command
 - Configure full application in docker-compose.yml

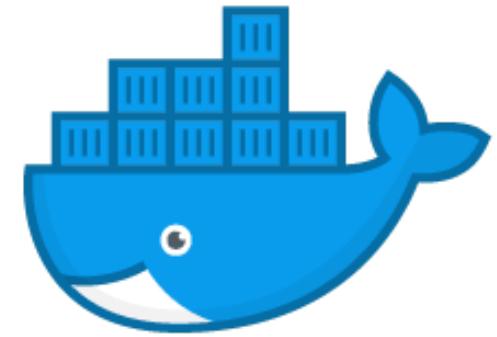
```
version: '3'
services:

psiturk:
  container_name: my-experiment
  build: .
  volumes:
    - ./exp:/exp
  tty: true
  stdin_open: true
  restart: unless-stopped

nginx:
  container_name: my-experiment-nginx
  image: nginx:latest
  ports:
    - 80:80
  volumes:
    - ./exp:/var/www/exp:ro
    - ./default.conf:/etc/nginx/conf.d/default.conf
  restart: unless-stopped

db:
  container_name: my-experiment-db
  image: mysql:5.7
  volumes:
    - ./data/db:/var/lib/mysql
  environment:
    MYSQL_ROOT_PASSWORD: mypassword
    MYSQL_DATABASE: participants
    MYSQL_USER: paxton
    MYSQL_PASSWORD: psiturk
  restart: unless-stopped

adminer:
  container_name: my-experiment-adminer
  image: adminer:latest
  ports:
    - 127.0.0.1:8080:8080
```



docker

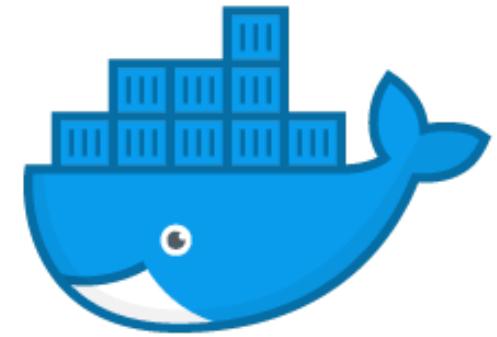
Container 1: psiTurk server

my_experiment/
└── exp/
 └── data/

```
version: '3'
services:

  psiturk:
    container_name: my-experiment
    build: .
    volumes:
      - ./exp:/exp
    tty: true
    stdin_open: true
    restart: unless-stopped

  nginx:
    container_name: my-experiment-nginx
    image: nginx:latest
    ports:
```



docker

docker-compose.yml

my_experiment/
└── exp/
 └── data/

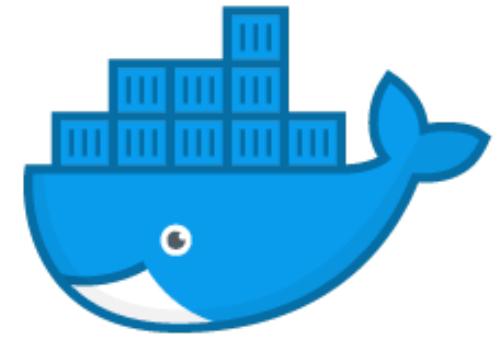
Container 1: psiTurk server

- Path to Dockerfile

```
version: '3'
services:

  psiturk:
    container_name: my-experiment
    build: .
    volumes:
      - ./exp:/exp
    tty: true
    stdin_open: true
    restart: unless-stopped

  nginx:
    container_name: my-experiment-nginx
    image: nginx:latest
    ports:
```



docker

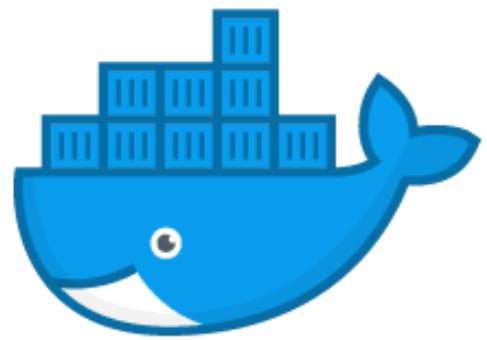
docker-compose.yml

```
my_experiment/  
└── exp/  
    └── data/
```

Container 1: psiTurk server

- Path to Dockerfile
- Mount the host directory `exp` to `/exp` in the container

```
version: '3'  
services:  
  
  psiturk:  
    container_name: my-experiment  
    build: .  
    volumes:  
      - ./exp:/exp  
    tty: true  
    stdin_open: true  
    restart: unless-stopped  
  
  nginx:  
    container_name: my-experiment-nginx  
    image: nginx:latest  
    ports:
```



docker

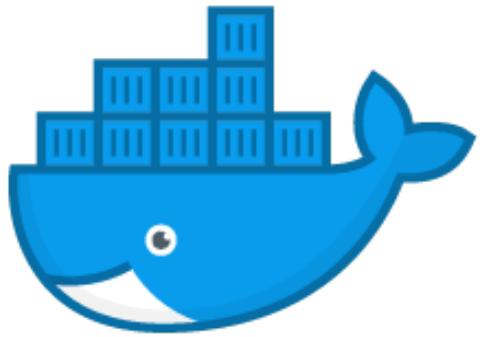
docker-compose.yml

```
my_experiment/  
└── exp/  
    └── data/
```

Container 1: psiTurk server

- Path to Dockerfile
- Mount the host directory `exp` to `/exp` in the container
- Allocate a pseudo-TTY and send `stdin` to the container (so we can run an interactive shell)

```
version: '3'  
services:  
  
  psiturk:  
    container_name: my-experiment  
    build: .  
    volumes:  
      - ./exp:/exp  
    tty: true  
    stdin_open: true  
    restart: unless-stopped  
  
  nginx:  
    container_name: my-experiment-nginx  
    image: nginx:latest  
    ports:
```



docker

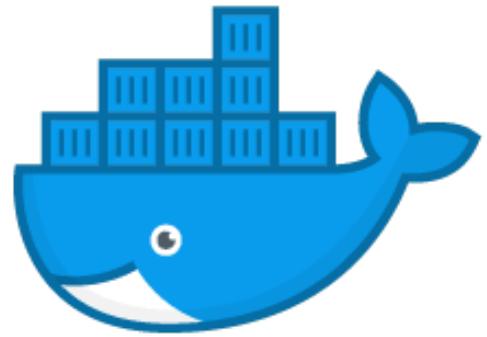
docker-compose.yml

```
my_experiment/  
└── exp/  
    └── data/
```

Container 1: psiTurk server

- Path to Dockerfile
- Mount the host directory `exp` to `/exp` in the container
- Allocate a pseudo-TTY and send stdin to the container (so we can run an interactive shell)
- If the container fails for some reason, restart it unless we explicitly stopped it.

```
version: '3'  
services:  
  
  psiturk:  
    container_name: my-experiment  
    build: .  
    volumes:  
      - ./exp:/exp  
    tty: true  
    stdin_open: true  
    restart: unless-stopped  
  
  nginx:  
    container_name: my-experiment-nginx  
    image: nginx:latest  
    ports:
```



docker

my_experiment/
└── exp/
 └── data/

docker-compose.yml

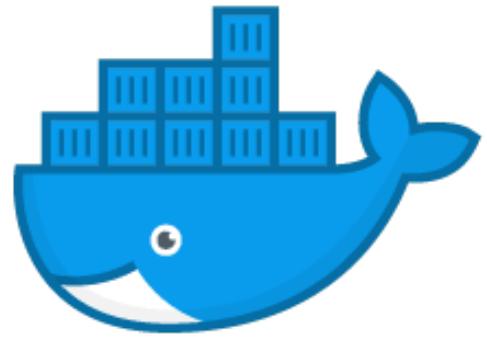
Container 2: NGINX

(reverse proxy server for load balancing)

```
build: .
volumes:
  - ./exp:/exp
tty: true
stdin_open: true
restart: unless-stopped
```

```
nginx:
  container_name: my-experiment-nginx
  image: nginx:latest
  ports:
    - 80:80
  volumes:
    - ./exp:/var/www/exp:ro
    - ./default.conf:/etc/nginx/conf.d/default.conf
  restart: unless-stopped
```

```
db:
  container_name: my-experiment-db
  image: mysql:5.7
  volumes:
    - ./data/db:/var/lib/mysql
  environment:
```

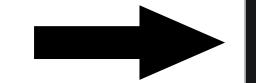


docker

Container 2: NGINX

(reverse proxy server for load balancing)

- Pull the nginx:latest image from Docker Hub

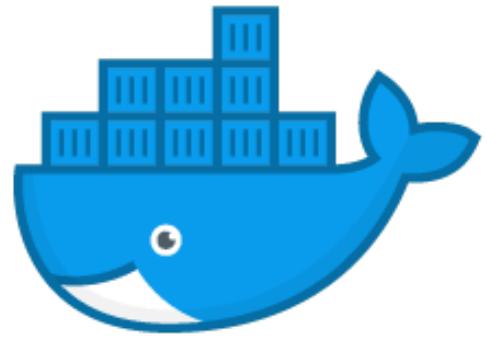


```
build: .
volumes:
- ./exp:/exp
tty: true
stdin_open: true
restart: unless-stopped

nginx:
container_name: my-experiment-nginx
image: nginx:latest
ports:
- 80:80
volumes:
- ./exp:/var/www/exp:ro
- ./default.conf:/etc/nginx/conf.d/default.conf
restart: unless-stopped

db:
container_name: my-experiment-db
image: mysql:5.7
volumes:
- ./data/db:/var/lib/mysql
environment:
```

```
my_experiment/
└── exp/
    └── data/
```



docker

Container 2: NGINX

(reverse proxy server for load balancing)

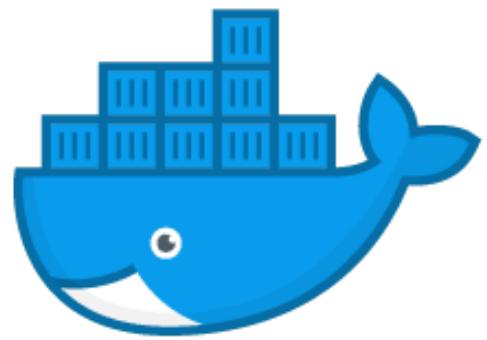
- Pull the nginx:latest image from Docker Hub
- Map host port 80 to container port 80

```
build: .
volumes:
- ./exp:/exp
tty: true
stdin_open: true
restart: unless-stopped

nginx:
container_name: my-experiment-nginx
image: nginx:latest
ports:
- 80:80
volumes:
- ./exp:/var/www/exp:ro
- ./default.conf:/etc/nginx/conf.d/default.conf
restart: unless-stopped

db:
container_name: my-experiment-db
image: mysql:5.7
volumes:
- ./data/db:/var/lib/mysql
environment:
```

```
my_experiment/
└── exp/
    └── data/
```



docker

Container 2: NGINX

(reverse proxy server for load balancing)

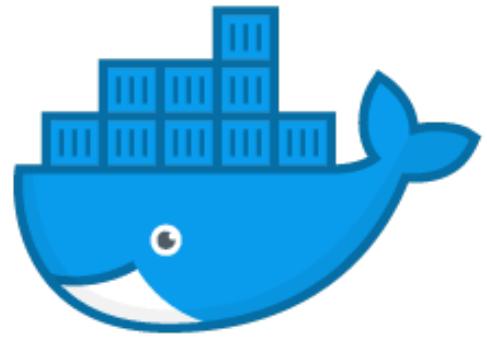
- Pull the nginx:latest image from Docker Hub
- Map host port 80 to container port 80
- Create two mount points:

```
my_experiment/
└── exp/
    └── data/
```

```
build: .
volumes:
- ./exp:/exp
tty: true
stdin_open: true
restart: unless-stopped

nginx:
container_name: my-experiment-nginx
image: nginx:latest
ports:
- 80:80
volumes:
- ./exp:/var/www/exp:ro
- ./default.conf:/etc/nginx/conf.d/default.conf
restart: unless-stopped

db:
container_name: my-experiment-db
image: mysql:5.7
volumes:
- ./data/db:/var/lib/mysql
environment:
```



docker

docker-compose.yml

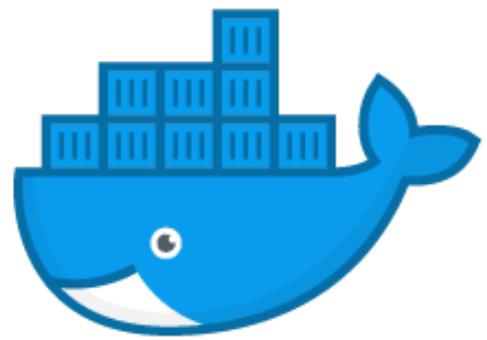
```
my_experiment/  
└── exp/  
    └── data/
```

Container 2: NGINX

(reverse proxy server for load balancing)

- Pull the nginx:latest image from Docker Hub
- Map host port 80 to container port 80
- Create two mount points:
 - host exp directory → NGINX web root

```
build: .  
volumes:  
  - ./exp:/exp  
tty: true  
stdin_open: true  
restart: unless-stopped  
  
nginx:  
  container_name: my-experiment-nginx  
  image: nginx:latest  
  ports:  
    - 80:80  
  volumes:  
    - ./exp:/var/www/exp:ro  
    - ./default.conf:/etc/nginx/conf.d/default.conf  
  restart: unless-stopped  
  
db:  
  container_name: my-experiment-db  
  image: mysql:5.7  
  volumes:  
    - ./data/db:/var/lib/mysql  
  environment:
```



docker

docker-compose.yml

my_experiment/
└── exp/
 └── data/

Container 2: NGINX

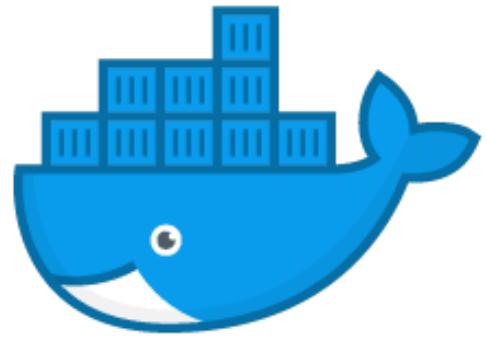
(reverse proxy server for load balancing)

- Pull the nginx:latest image from Docker Hub
- Map host port 80 to container port 80
- Create two mount points:
 - host exp directory → NGINX web root
 - NGINX config file → expected location

```
build: .
volumes:
  - ./exp:/exp
tty: true
stdin_open: true
restart: unless-stopped

nginx:
  container_name: my-experiment-nginx
  image: nginx:latest
  ports:
    - 80:80
  volumes:
    - ./exp:/var/www/exp:ro
    - ./default.conf:/etc/nginx/conf.d/default.conf
  restart: unless-stopped

db:
  container_name: my-experiment-db
  image: mysql:5.7
  volumes:
    - ./data/db:/var/lib/mysql
  environment:
```



docker

Container 3: MySQL database

(greater concurrency than default SQLite DB)

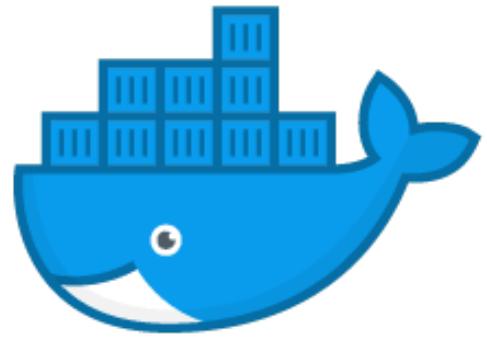
docker-compose.yml

my_experiment/
└── exp/
 └── data/

```
      - 80:80
volumes:
  - ./exp:/var/www/exp:ro
  - ./default.conf:/etc/nginx/conf.d/default.conf
restart: unless-stopped
```

```
db:
  container_name: my-experiment-db
  image: mysql:5.7
  volumes:
    - ./data/db:/var/lib/mysql
  environment:
    MYSQL_ROOT_PASSWORD: mypassword
    MYSQL_DATABASE: participants
    MYSQL_USER: paxton
    MYSQL_PASSWORD: psiturk
  restart: unless-stopped
```

```
adminer:
  container_name: my-experiment-adminer
  image: adminer:latest
  ports:
    - 127.0.0.1:8080:8080
```



docker

docker-compose.yml

my_experiment/
└── exp/
 └── data/

Container 3: MySQL database

(greater concurrency than default SQLite DB)

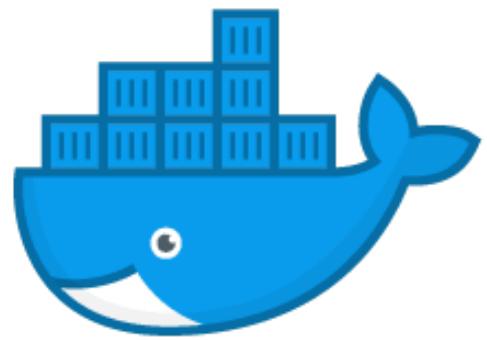
- Pull the mysql:5.7 image from Docker Hub



```
- 80:80
volumes:
- ./exp:/var/www/exp:ro
- ./default.conf:/etc/nginx/conf.d/default.conf
restart: unless-stopped

db:
  container_name: my-experiment-db
  image: mysql:5.7
  volumes:
    - ./data/db:/var/lib/mysql
  environment:
    MYSQL_ROOT_PASSWORD: mypassword
    MYSQL_DATABASE: participants
    MYSQL_USER: paxton
    MYSQL_PASSWORD: psiturk
  restart: unless-stopped

adminer:
  container_name: my-experiment-adminer
  image: adminer:latest
  ports:
    - 127.0.0.1:8080:8080
```



docker

Container 3: MySQL database

(greater concurrency than default SQLite DB)

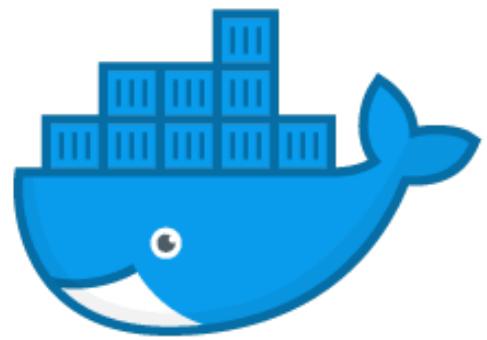
- Pull the mysql:5.7 image from Docker Hub
- Mount data/db directory on host to /var/lib/mysql in container

```
  - 80:80
volumes:
  - ./exp:/var/www/exp:ro
  - ./default.conf:/etc/nginx/conf.d/default.conf
restart: unless-stopped

db:
  container_name: my-experiment-db
  image: mysql:5.7
  volumes:
    - ./data/db:/var/lib/mysql
  environment:
    MYSQL_ROOT_PASSWORD: mypassword
    MYSQL_DATABASE: participants
    MYSQL_USER: paxton
    MYSQL_PASSWORD: psiturk
  restart: unless-stopped

adminer:
  container_name: my-experiment-adminer
  image: adminer:latest
  ports:
    - 127.0.0.1:8080:8080
```

```
my_experiment/
└── exp/
    └── data/
```



docker

docker-compose.yml

my_experiment/
└── exp/
 └── data/

Container 3: MySQL database

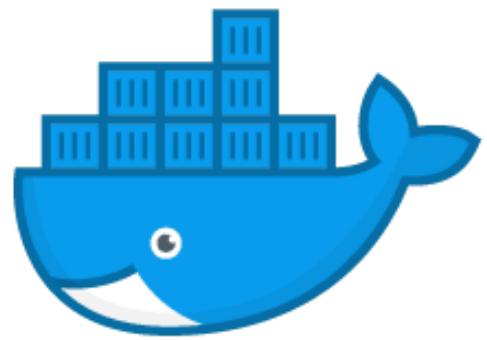
(greater concurrency than default SQLite DB)

- Pull the mysql:5.7 image from Docker Hub
- Mount data/db directory on host to /var/lib/mysql in container
- Set container environment variables:

```
- 80:80
volumes:
- ./exp:/var/www/exp:ro
- ./default.conf:/etc/nginx/conf.d/default.conf
restart: unless-stopped

db:
  container_name: my-experiment-db
  image: mysql:5.7
  volumes:
    - ./data/db:/var/lib/mysql
  environment:
    MYSQL_ROOT_PASSWORD: mypassword
    MYSQL_DATABASE: participants
    MYSQL_USER: paxton
    MYSQL_PASSWORD: psitruk
  restart: unless-stopped
```

```
adminer:
  container_name: my-experiment-adminer
  image: adminer:latest
  ports:
    - 127.0.0.1:8080:8080
```



docker

docker-compose.yml

my_experiment/
└── exp/
 └── data/

Container 3: MySQL database

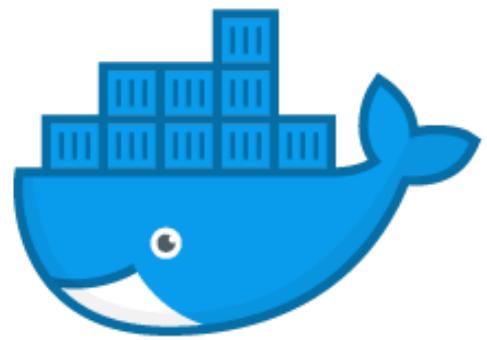
(greater concurrency than default SQLite DB)

- Pull the mysql:5.7 image from Docker Hub
- Mount data/db directory on host to /var/lib/mysql in container
- Set container environment variables:
 - Password for MySQL root user (required)

```
- 80:80
volumes:
- ./exp:/var/www/exp:ro
- ./default.conf:/etc/nginx/conf.d/default.conf
restart: unless-stopped

db:
  container_name: my-experiment-db
  image: mysql:5.7
  volumes:
    - ./data/db:/var/lib/mysql
  environment:
    MYSQL_ROOT_PASSWORD: mypassword
    MYSQL_DATABASE: participants
    MYSQL_USER: paxton
    MYSQL_PASSWORD: psiturk
  restart: unless-stopped

adminer:
  container_name: my-experiment-adminer
  image: adminer:latest
  ports:
    - 127.0.0.1:8080:8080
```



docker

docker-compose.yml

my_experiment/
└── exp/
 └── data/

Container 3: MySQL database

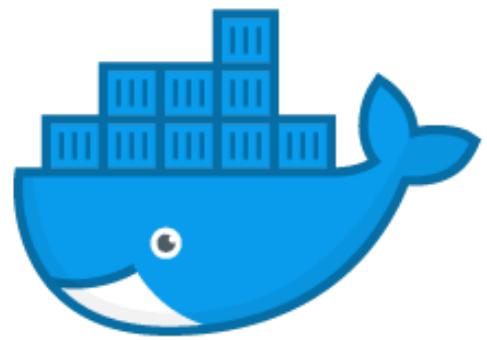
(greater concurrency than default SQLite DB)

- Pull the mysql:5.7 image from Docker Hub
- Mount data/db directory on host to /var/lib/mysql in container
- Set container environment variables:
 - Password for MySQL root user (required)
 - Name of DB created in the container

```
  - 80:80
volumes:
  - ./exp:/var/www/exp:ro
  - ./default.conf:/etc/nginx/conf.d/default.conf
restart: unless-stopped

db:
  container_name: my-experiment-db
  image: mysql:5.7
  volumes:
    - ./data/db:/var/lib/mysql
  environment:
    MYSQL_ROOT_PASSWORD: mypassword
    MYSQL_DATABASE: participants
    MYSQL_USER: paxton
    MYSQL_PASSWORD: psiturk
  restart: unless-stopped

adminer:
  container_name: my-experiment-adminer
  image: adminer:latest
  ports:
    - 127.0.0.1:8080:8080
```



docker

Container 3: MySQL database

(greater concurrency than default SQLite DB)

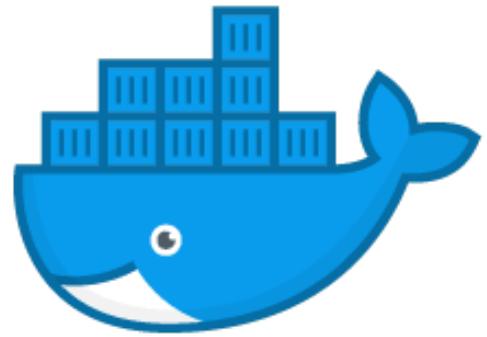
- Pull the mysql:5.7 image from Docker Hub
- Mount data/db directory on host to /var/lib/mysql in container
- Set container environment variables:
 - Password for MySQL root user (required)
 - Name of DB created in the container
 - Username/password for psiTurk to write to DB

```
      - 80:80
volumes:
  - ./exp:/var/www/exp:ro
  - ./default.conf:/etc/nginx/conf.d/default.conf
restart: unless-stopped

db:
  container_name: my-experiment-db
  image: mysql:5.7
  volumes:
    - ./data/db:/var/lib/mysql
  environment:
    MYSQL_ROOT_PASSWORD: mypassword
    MYSQL_DATABASE: participants
    MYSQL_USER: paxton
    MYSQL_PASSWORD: psiturk
  restart: unless-stopped

adminer:
  container_name: my-experiment-adminer
  image: adminer:latest
  ports:
    - 127.0.0.1:8080:8080
```

```
my_experiment/
└── exp/
    └── data/
```



docker

Container 4: Adminer

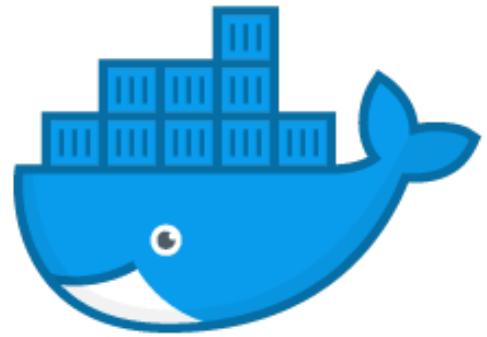
(PHP app for viewing/downloading data)

docker-compose.yml

```
my_experiment/  
└── exp/  
    └── data/
```

```
volumes:  
  - ./data/db:/var/lib/mysql  
environment:  
  MYSQL_ROOT_PASSWORD: mypassword  
  MYSQL_DATABASE: participants  
  MYSQL_USER: paxton  
  MYSQL_PASSWORD: psiturk  
restart: unless-stopped
```

```
adminer:  
  container_name: my-experiment-adminer  
  image: adminer:latest  
  ports:  
    - 127.0.0.1:8080:8080
```



docker

my_experiment/
└── exp/
 └── data/

docker-compose.yml

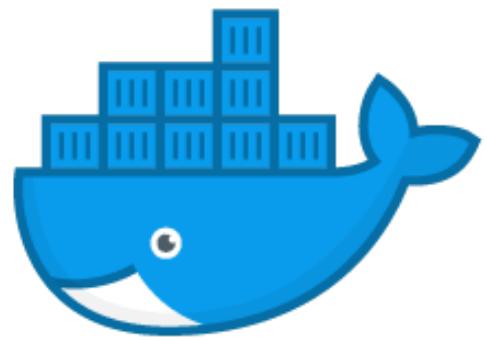
Container 4: Adminer

(PHP app for viewing/downloading data)

- Pull the adminer:latest image from Docker Hub



```
volumes:  
  - ./data/db:/var/lib/mysql  
environment:  
  MYSQL_ROOT_PASSWORD: mypassword  
  MYSQL_DATABASE: participants  
  MYSQL_USER: paxton  
  MYSQL_PASSWORD: psiturk  
restart: unless-stopped  
  
adminer:  
  container_name: my-experiment-adminer  
  image: adminer:latest  
  ports:  
    - 127.0.0.1:8080:8080
```



docker

Container 4: Adminer

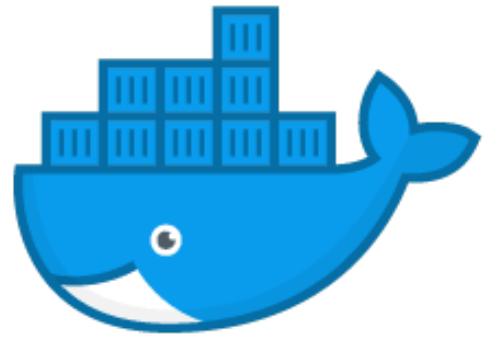
(PHP app for viewing/downloading data)

- Pull the adminer:latest image from Docker Hub
- Map host port 8080 to container port 8080
(allows us to access Adminer via a web browser)

```
version: '2'
  - ./data/db:/var/lib/mysql
environment:
  MYSQL_ROOT_PASSWORD: mypassword
  MYSQL_DATABASE: participants
  MYSQL_USER: paxton
  MYSQL_PASSWORD: psiturk
restart: unless-stopped

adminer:
  container_name: my-experiment-adminer
  image: adminer:latest
  ports:
    - 127.0.0.1:8080:8080
```

my_experiment/
└── exp/
 └── data/



docker

docker-compose.yml

my_experiment/
└── exp/
 └── data/

```
version: '3'
services:

  psiturk:
    container_name: my-experiment
    build: .
    volumes:
      - ./exp:/exp
    tty: true
    stdin_open: true
    restart: unless-stopped

  nginx:
    container_name: my-experiment-nginx
    image: nginx:latest
    ports:
      - 80:80
    volumes:
      - ./exp:/var/www/exp:ro
      - ./default.conf:/etc/nginx/conf.d/default.conf
    restart: unless-stopped

  db:
    container_name: my-experiment-db
    image: mysql:5.7
    volumes:
      - ./data/db:/var/lib/mysql
    environment:
      MYSQL_ROOT_PASSWORD: mypassword
      MYSQL_DATABASE: participants
      MYSQL_USER: paxton
      MYSQL_PASSWORD: psiturk
    restart: unless-stopped

  adminer:
    container_name: my-experiment-adminer
    image: adminer:latest
    ports:
      - 127.0.0.1:8080:8080
```

Experiment structure

```
my_experiment/  
└── exp/  
└── data/
```

Experiment structure

```
my_experiment/
└── exp/
└── data/
└── Dockerfile
└── docker-compose.yml
└── default.conf
```

Experiment structure

```
server {
    listen 80 default_server;
    server_name localhost;
    root /var/www/exp/templates;

    location / {
        try_files $uri @proxy_to_psiturk;
    }

    location @proxy_to_psiturk {
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;
        proxy_redirect off;
        proxy_pass http://psiturk:22362;
    }
}
```

Experiment structure

```
my_experiment/
└── exp/
└── data/
└── Dockerfile
└── docker-compose.yml
└── default.conf
```

Experiment structure

```
my_experiment/
├── exp/
├── data/
│   └── db/
└── Dockerfile
    docker-compose.yml
    default.conf
```



What is psiTurk?

```
my_experiment/
  └── exp/
  └── data/
  └── Dockerfile
  └── docker-compose.yml
  └── default.conf
```



What is psiTurk?

```
my_experiment/
  └── exp/
  └── data/
  └── Dockerfile
  └── docker-compose.yml
  └── default.conf
```

- Framework for creating & running browser-based behavioral experiments



What is psiTurk?

```
my_experiment/
  └── exp/
  └── data/
  └── Dockerfile
  └── docker-compose.yml
  └── default.conf
```

- Framework for creating & running browser-based behavioral experiments
- Flask app, Python web server, JavaScript library



What is psiTurk?

```
my_experiment/
  └── exp/
  └── data/
  └── Dockerfile
  └── docker-compose.yml
  └── default.conf
```

- Framework for creating & running browser-based behavioral experiments
- Flask app, Python web server, JavaScript library
- Interfaces with MTurk



Why use psiTurk?

```
my_experiment/
  └── exp/
  └── data/
  └── Dockerfile
  └── docker-compose.yml
  └── default.conf
```



Why use psiTurk?

```
my_experiment/
  └── exp/
  └── data/
  └── Dockerfile
  └── docker-compose.yml
  └── default.conf
```

- MTurk experiments must be web-based (no PsychoPy, Psychtoolbox, etc.)



Why use psiTurk?

```
my_experiment/
  └── exp/
  └── data/
  └── Dockerfile
  └── docker-compose.yml
  └── default.conf
```

- MTurk experiments must be web-based (no PsychoPy, Psychtoolbox, etc.)
- Wraps and simplifies every part of the process
 - Configuring & running web server
 - Error logging
 - Interfacing with reverse proxy
 - Authenticating with AWS
 - Hosting ad on secure server
 - Testing in MTurk sandbox
 - Configuring & posting HIT
 - Setting requirements/restrictions
 - Providing consent form
 - Saving data
 - Approving/rejecting HITs
 - Expiring/deleting HITs
 - Checking account balance
 - Compensating participants
 - Granting bonuses



Why use psiTurk?

```
my_experiment/
  └── exp/
  └── data/
  └── Dockerfile
  └── docker-compose.yml
  └── default.conf
```

- MTurk experiments must be web-based (no PsychoPy, Psychtoolbox, etc.)
- Wraps and simplifies every part of the process
 - Configuring & running web server
 - Error logging
 - Interfacing with reverse proxy
 - Authenticating with AWS
 - Hosting ad on secure server
 - Testing in MTurk sandbox
 - Configuring & posting HIT
 - Setting requirements/restrictions
 - Providing consent form
 - Saving data
 - Approving/rejecting HITs
 - Expiring/deleting HITs
 - Checking account balance
 - Compensating participants
 - Granting bonuses
- Run experiment locally with no changes



Why use psiTurk?

```
my_experiment/
  └── exp/
  └── data/
  └── Dockerfile
  └── docker-compose.yml
  └── default.conf
```

- MTurk experiments must be web-based (no PsychoPy, Psychtoolbox, etc.)
- Wraps and simplifies every part of the process
 - Configuring & running web server
 - Error logging
 - Interfacing with reverse proxy
 - Authenticating with AWS
 - Hosting ad on secure server
 - Testing in MTurk sandbox
 - Configuring & posting HIT
 - Setting requirements/restrictions
 - Providing consent form
 - Saving data
 - Approving/rejecting HITs
 - Expiring/deleting HITs
 - Checking account balance
 - Compensating participants
 - Granting bonuses
- Run experiment locally with no changes
 - Compare/validate MTurk data



Why use psiTurk?

```
my_experiment/
  └── exp/
  └── data/
  └── Dockerfile
  └── docker-compose.yml
  └── default.conf
```

- MTurk experiments must be web-based (no PsychoPy, Psychtoolbox, etc.)
- Wraps and simplifies every part of the process
 - Configuring & running web server
 - Error logging
 - Interfacing with reverse proxy
 - Authenticating with AWS
 - Hosting ad on secure server
 - Testing in MTurk sandbox
 - Configuring & posting HIT
 - Setting requirements/restrictions
 - Providing consent form
 - Saving data
 - Approving/rejecting HITs
 - Expiring/deleting HITs
 - Checking account balance
 - Compensating participants
 - Granting bonuses
- Run experiment locally with no changes
 - Compare/validate MTurk data
 - Fully in-person studies



Why use psiTurk?

```
my_experiment/
  └── exp/
  └── data/
  └── Dockerfile
  └── docker-compose.yml
  └── default.conf
```

- MTurk experiments must be web-based (no PsychoPy, Psychtoolbox, etc.)
- Wraps and simplifies every part of the process
 - Configuring & running web server
 - Error logging
 - Interfacing with reverse proxy
 - Authenticating with AWS
 - Hosting ad on secure server
 - Testing in MTurk sandbox
 - Configuring & posting HIT
 - Setting requirements/restrictions
 - Providing consent form
 - Saving data
 - Approving/rejecting HITs
 - Expiring/deleting HITs
 - Checking account balance
 - Compensating participants
 - Granting bonuses
- Run experiment locally with no changes
 - Compare/validate MTurk data
 - Fully in-person studies
- Handles heavy stimuli/data other options can't



Anatomy of a Psiturk experiment

```
my_experiment/
  └── exp/
  └── data/
  └── Dockerfile
  └── docker-compose.yml
  └── default.conf
```



Anatomy of a Psiturk experiment

```
my_experiment/
  └── exp/
  └── data/
  └── Dockerfile
  └── docker-compose.yml
  └── default.conf
```



Anatomy of a Psiturk experiment

```
my_experiment/
  └── exp/
  └── data/
  └── Dockerfile
  └── docker-compose.yml
  └── default.conf
```



Anatomy of a Psiturk experiment

```
my_experiment/  
  └── exp/  
  └── data/  
  └── Dockerfile  
  └── docker-compose.yml  
  └── default.conf
```

exp/



Anatomy of a PsiTurk experiment

- psiTurk and AWS credentials

psiturk.readthedocs.io/en/latest/amt_setup.html

psiturk.readthedocs.io/en/latest/psiturk_org_setup.html

exp/
 └ .psiturkconfig

```
my_experiment/
  └── exp/
  └── data/
  └── Dockerfile
  └── docker-compose.yml
  └── default.conf
```



Anatomy of a PsiTurk experiment

```
my_experiment/
  └── exp/
  └── data/
  └── Dockerfile
  └── docker-compose.yml
  └── default.conf
```

- psiTurk and AWS credentials
 - psiturk.readthedocs.io/en/latest/amt_setup.html
 - psiturk.readthedocs.io/en/latest/psiturk_org_setup.html
- Full psiTurk & Mturk configuration

```
exp/
  └── .psiturkconfig
  └── config.txt
```



Anatomy of a PsiTurk experiment

```
my_experiment/
  └── exp/
  └── data/
  └── Dockerfile
  └── docker-compose.yml
  └── default.conf
```

- psiTurk and AWS credentials
 - psiturk.readthedocs.io/en/latest/amt_setup.html
 - psiturk.readthedocs.io/en/latest/psiturk_org_setup.html
- Full psiTurk & Mturk configuration
- Define custom URLs for complex designs
(optional)

```
exp/
  └── .psiturkconfig
  └── config.txt
  └── custom.py
```



Anatomy of a PsiTurk experiment

```
my_experiment/
  └── exp/
  └── data/
  └── Dockerfile
  └── docker-compose.yml
  └── default.conf
```

- psiTurk and AWS credentials
 - psiturk.readthedocs.io/en/latest/amt_setup.html
 - psiturk.readthedocs.io/en/latest/psiturk_org_setup.html
- Full psiTurk & Mturk configuration
- Define custom URLs for complex designs
(optional)
- Static files

```
exp/
  └── .psiturkconfig
  └── config.txt
  └── custom.py
  └── static/
```



Anatomy of a PsiTurk experiment

```
my_experiment/
  └── exp/
  └── data/
  └── Dockerfile
  └── docker-compose.yml
  └── default.conf
```

- psiTurk and AWS credentials
 - psiturk.readthedocs.io/en/latest/amt_setup.html
 - psiturk.readthedocs.io/en/latest/psiturk_org_setup.html
- Full psiTurk & Mturk configuration
- Define custom URLs for complex designs (optional)
- Static files
- Bare-bones HTML templates

```
exp/
  └── .psiturkconfig
  └── config.txt
  └── custom.py
  └── static/
  └── templates/
```



PSITURK

config.txt

```
● ● ●

[HIT Configuration]
title = Example MTurk experiment
description = Judge the color of a series of words.
amt_keywords = Perception, Psychology
organization_name = Dartmouth College
browser_exclude_rule = MSIE, mobile, tablet
contact_email_on_error = Paxton.C.Fitzpatrick@Dartmouth.edu
lifetime = 1
ad_group = Example Experiments
allow_repeats = false
us_only = true
approve_requirement = 95
number_hits_approved = 0
require_master_workers = false

[Database Parameters]
database_url = mysql://paxton:psiturk@db:3306/participants
table_name = turkdemo

[Server Parameters]
host = 0.0.0.0
port = 22362
cutoff_time = 30
logfile = server.log
loglevel = 2
debug = true
threads = auto
adserver_revproxy_host = 127.0.0.1
adserver_revproxy_port = 80

[Task Parameters]
experiment_code_version = 1.0
num_conds = 1
num_counters = 1

[Shell Parameters]
launch_in_sandbox_mode = true
use_psiturk_ad_server = true
ad_location = false
```

```
exp/
└── .psiturkconfig
    ├── config.txt
    └── custom.py
    └── static/
        └── templates/
```



config.txt

```
exp/
  └── .psiturkconfig
  └── config.txt
  └── custom.py
  └── static/
  └── templates/
```

HIT Configuration



```
[HIT Configuration]
title = Example MTurk experiment
description = Judge the color of a series of words.
amt_keywords = Perception, Psychology
organization_name = Dartmouth College
browser_exclude_rule = MSIE, mobile, tablet
contact_email_on_error = Paxton.C.Fitzpatrick@Dartmouth.edu
lifetime = 1
ad_group = Example Experiments
allow_repeats = false
us_only = true
approve_requirement = 95
number_hits_approved = 0
require_master_workers = false
```

```
[Database Parameters]
database_url = mysql://paxton:psiturk@db:3306/participants
table_name = turkdemo
```

```
[Server Parameters]
```



config.txt

```
exp/
  └── .psiturkconfig
  └── config.txt
  └── custom.py
  └── static/
  └── templates/
```

HIT Configuration: participant eligibility



```
[HIT Configuration]
title = Example MTurk experiment
description = Judge the color of a series of words.
amt_keywords = Perception, Psychology
organization_name = Dartmouth College
browser_exclude_rule = MSIE, mobile, tablet
contact_email_on_error = Paxton.C.Fitzpatrick@Dartmouth.edu
lifetime = 1
ad_group = Example Experiments
allow_repeats = false
us_only = true
approve_requirement = 95
number_hits_approved = 0
require_master_workers = false
```

```
[Database Parameters]
database_url = mysql://paxton:psiturk@db:3306/participants
table_name = turkdemo
```

```
[Server Parameters]
```



config.txt

```
exp/
  └── .psiturkconfig
  └── config.txt
  └── custom.py
  └── static/
  └── templates/
```

HIT Configuration: participant eligibility

- Ineligible browsers/devices



```
● ● ●

[HIT Configuration]
title = Example MTurk experiment
description = Judge the color of a series of words.
amt_keywords = Perception, Psychology
organization_name = Dartmouth College
browser_exclude_rule = MSIE, mobile, tablet
contact_email_on_error = Paxton.C.Fitzpatrick@Dartmouth.edu
lifetime = 1
ad_group = Example Experiments
allow_repeats = false
us_only = true
approve_requirement = 95
number_hits_approved = 0
require_master_workers = false
```

```
[Database Parameters]
database_url = mysql://paxton:psiturk@db:3306/participants
table_name = turkdemo
```

```
[Server Parameters]
```

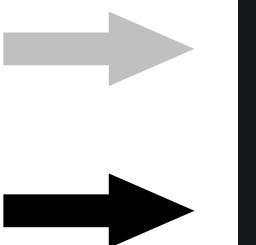


config.txt

```
exp/
  └── .psiturkconfig
  └── config.txt
  └── custom.py
  └── static/
  └── templates/
```

HIT Configuration: participant eligibility

- Ineligible browsers/devices
- How long HIT listing is visible



```
● ● ●

[HIT Configuration]
title = Example MTurk experiment
description = Judge the color of a series of words.
amt_keywords = Perception, Psychology
organization_name = Dartmouth College
browser_exclude_rule = MSIE, mobile, tablet
contact_email_on_error = Paxton.C.Fitzpatrick@Dartmouth.edu
lifetime = 1
ad_group = Example Experiments
allow_repeats = false
us_only = true
approve_requirement = 95
number_hits_approved = 0
require_master_workers = false
```

```
[Database Parameters]
database_url = mysql://paxton:psiturk@db:3306/participants
table_name = turkdemo
```

```
[Server Parameters]
```

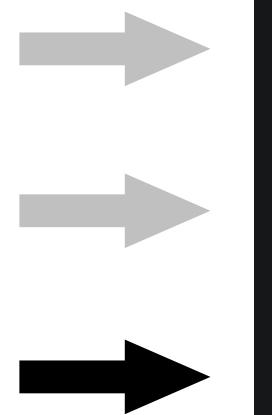


config.txt

```
exp/
  └── .psiturkconfig
  └── config.txt
  └── custom.py
  └── static/
  └── templates/
```

HIT Configuration: participant eligibility

- Ineligible browsers/devices
- How long HIT listing is visible
- Participate once or multiple times



```
● ● ●

[HIT Configuration]
title = Example MTurk experiment
description = Judge the color of a series of words.
amt_keywords = Perception, Psychology
organization_name = Dartmouth College
browser_exclude_rule = MSIE, mobile, tablet
contact_email_on_error = Paxton.C.Fitzpatrick@Dartmouth.edu
lifetime = 1
ad_group = Example Experiments
allow_repeats = false
us_only = true
approve_requirement = 95
number_hits_approved = 0
require_master_workers = false
```

```
[Database Parameters]
database_url = mysql://paxton:psiturk@db:3306/participants
table_name = turkdemo
```

```
[Server Parameters]
```

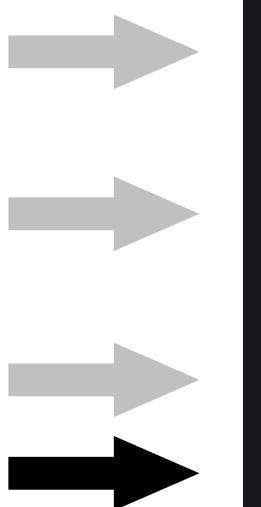


config.txt

```
exp/
  └── .psiturkconfig
  └── config.txt
  └── custom.py
  └── static/
  └── templates/
```

HIT Configuration: participant eligibility

- Ineligible browsers/devices
- How long HIT listing is visible
- Participate once or multiple times
- Restrict to US-based participants



```
● ● ●

[HIT Configuration]
title = Example MTurk experiment
description = Judge the color of a series of words.
amt_keywords = Perception, Psychology
organization_name = Dartmouth College
browser_exclude_rule = MSIE, mobile, tablet
contact_email_on_error = Paxton.C.Fitzpatrick@Dartmouth.edu
lifetime = 1
ad_group = Example Experiments
allow_repeats = false
us_only = true
approve_requirement = 95
number_hits_approved = 0
require_master_workers = false
```

```
[Database Parameters]
database_url = mysql://paxton:psiturk@db:3306/participants
table_name = turkdemo
```

```
[Server Parameters]
```

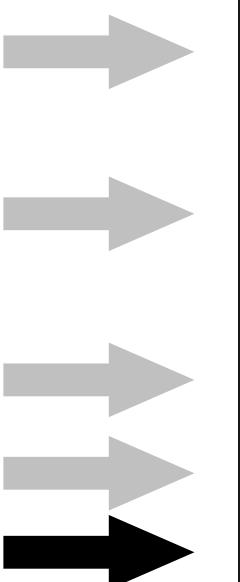


config.txt

```
exp/
  └── .psiturkconfig
  └── config.txt
  └── custom.py
  └── static/
  └── templates/
```

HIT Configuration: participant eligibility

- Ineligible browsers/devices
- How long HIT listing is visible
- Participate once or multiple times
- Restrict to US-based participants
- Percentage of previous HITs approved



```
● ● ●  
[HIT Configuration]  
title = Example MTurk experiment  
description = Judge the color of a series of words.  
amt_keywords = Perception, Psychology  
organization_name = Dartmouth College  
browser_exclude_rule = MSIE, mobile, tablet  
contact_email_on_error = Paxton.C.Fitzpatrick@Dartmouth.edu  
lifetime = 1  
ad_group = Example Experiments  
allow_repeats = false  
us_only = true  
approve_requirement = 95  
number_hits_approved = 0  
require_master_workers = false
```

```
[Database Parameters]  
database_url = mysql://paxton:psiturk@db:3306/participants  
table_name = turkdemo
```

```
[Server Parameters]
```

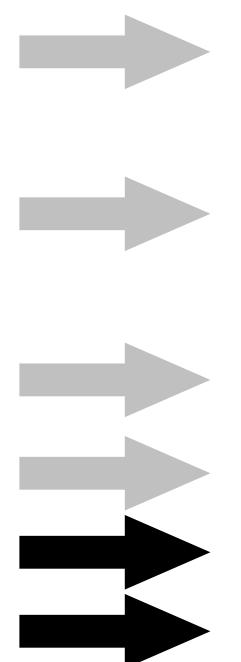


config.txt

```
exp/
  └── .psiturkconfig
  └── config.txt
  └── custom.py
  └── static/
  └── templates/
```

HIT Configuration: participant eligibility

- Ineligible browsers/devices
- How long HIT listing is visible
- Participate once or multiple times
- Restrict to US-based participants
- Percentage of previous HITs approved
- Number of previous HITs approved



```
● ● ●

[HIT Configuration]
title = Example MTurk experiment
description = Judge the color of a series of words.
amt_keywords = Perception, Psychology
organization_name = Dartmouth College
browser_exclude_rule = MSIE, mobile, tablet
contact_email_on_error = Paxton.C.Fitzpatrick@Dartmouth.edu
lifetime = 1
ad_group = Example Experiments
allow_repeats = false
us_only = true
approve_requirement = 95
number_hits_approved = 0
require_master_workers = false
```

```
[Database Parameters]
database_url = mysql://paxton:psiturk@db:3306/participants
table_name = turkdemo
```

```
[Server Parameters]
```



config.txt

```
exp/
  └── .psiturkconfig
  └── config.txt
  └── custom.py
  └── static/
  └── templates/
```

HIT Configuration: participant eligibility

- Ineligible browsers/devices
- How long HIT listing is visible
- Participate once or multiple times
- Restrict to US-based participants
- Percentage of previous HITs approved
- Number of previous HITs approved
- “Master worker” designation by MTurk (5% compensation fee)

```
● ● ●

[HIT Configuration]
title = Example MTurk experiment
description = Judge the color of a series of words.
amt_keywords = Perception, Psychology
organization_name = Dartmouth College
browser_exclude_rule = MSIE, mobile, tablet
contact_email_on_error = Paxton.C.Fitzpatrick@Dartmouth.edu
lifetime = 1
ad_group = Example Experiments
allow_repeats = false
us_only = true
approve_requirement = 95
number_hits_approved = 0
require_master_workers = false

[Database Parameters]
database_url = mysql://paxton:psiturk@db:3306/participants
table_name = turkdemo

[Server Parameters]
host = 127.0.0.1
port = 8000
```



config.txt

```
exp/
  └── .psiturkconfig
  └── config.txt
  └── custom.py
  └── static/
  └── templates/
```

Database Parameters

```
allow_repeats = false
us_only = true
approve_requirement = 95
number_hits_approved = 0
require_master_workers = false
```

```
[Database Parameters]
database_url = mysql://paxton:psiturk@db:3306/participants
table_name = turkdemo
```

```
[Server Parameters]
host = 0.0.0.0
port = 22362
cutoff_time = 30
logfile = server.log
```



config.txt

```
exp/
  └── .psiturkconfig
  └── config.txt
  └── custom.py
  └── static/
  └── templates/
```

Database Parameters

- SQLAlchemy connection string
(fields from docker-compose.yml)



```
db:
  container_name: my-experiment-db
  image: mysql:5.7
  volumes:
    - ./data/db:/var/lib/mysql
  environment:
    MYSQL_ROOT_PASSWORD: mypassword
    MYSQL_DATABASE: participants
    MYSQL_USER: paxton
    MYSQL_PASSWORD: psiturk
  restart: unless-stopped
```

```
allow_repeats = false
us_only = true
approve_requirement = 95
number_hits_approved = 0
require_master_workers = false

[Database Parameters]
database_url = mysql://paxton:psiturk@db:3306/participants
table_name = turkdemo

[Server Parameters]
host = 0.0.0.0
port = 22362
cutoff_time = 30
logfile = server.log
```

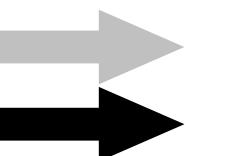


config.txt

```
exp/
  └── .psiturkconfig
  └── config.txt
  └── custom.py
  └── static/
  └── templates/
```

Database Parameters

- SQLAlchemy connection string
(fields from docker-compose.yml)
- Table name (arbitrary)



```
db:  
  container_name: my-experiment-db  
  image: mysql:5.7  
  volumes:  
    - ./data/db:/var/lib/mysql  
  environment:  
    MYSQL_ROOT_PASSWORD: mypassword  
    MYSQL_DATABASE: participants  
    MYSQL_USER: paxton  
    MYSQL_PASSWORD: psiturk  
  restart: unless-stopped
```

```
allow_repeats = false  
us_only = true  
approve_requirement = 95  
number_hits_approved = 0  
require_master_workers = false  
  
[Database Parameters]  
database_url = mysql://paxton:psiturk@db:3306/participants  
table_name = turkdemo
```

```
[Server Parameters]  
host = 0.0.0.0  
port = 22362  
cutoff_time = 30  
logfile = server.log
```



PSITURK

Server Parameters

config.txt

```
require_master_workers = false
```

```
[Database Parameters]
```

```
database_url = mysql://paxton:psiturk@db:3306/participants
table_name = turkdemo
```

```
[Server Parameters]
```

```
host = 0.0.0.0
port = 22362
cutoff_time = 30
logfile = server.log
loglevel = 2
debug = true
threads = auto
adserver_revproxy_host = 127.0.0.1
adserver_revproxy_port = 80
```

```
[Task Parameters]
```

```
experiment_code_version = 1.0
numconds = 1
num_counters = 1
```

```
exp/
└── .psiturkconfig
    └── config.txt
    └── custom.py
    └── static/
        └── templates/
```



PSITURK

Server Parameters

- Correspond to `default.conf`

● ● ●

```
server {  
    listen 80 default_server;  
    server_name localhost;  
    root /var/www/exp/templates;  
  
    location / {  
        try_files $uri @proxy_to_psiturk;  
    }  
  
    location @proxy_to_psiturk {  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
        proxy_set_header Host $http_host;  
        proxy_redirect off;  
        proxy_pass http://psiturk:22362;  
    }  
}
```

config.txt

```
require_master_workers = false
```

[Database Parameters]

```
database_url = mysql://paxton:psiturk@db:3306/participants  
table_name = turkdemo
```

[Server Parameters]

```
host = 0.0.0.0  
port = 22362  
cutoff_time = 30  
logfile = server.log  
loglevel = 2  
debug = true  
threads = auto  
adserver_revproxy_host = 127.0.0.1  
adserver_revproxy_port = 80
```

[Task Parameters]

```
experiment_code_version = 1.0  
num_conds = 1  
num_counters = 1
```

```
exp/  
  └── .psiturkconfig  
  └── config.txt  
  └── custom.py  
  └── static/  
  └── templates/
```



config.txt

```
exp/
  └── .psiturkconfig
  └── config.txt
  └── custom.py
  └── static/
  └── templates/
```

Server Parameters

- Correspond to default.conf

```
require_master_workers = false

[Database Parameters]
database_url = mysql://paxton:psiturk@db:3306/participants
table_name = turkdemo

[Server Parameters]
host = 0.0.0.0
port = 22362
cutoff_time = 30
logfile = server.log
loglevel = 2
debug = true
threads = auto
adserver_revproxy_host = 127.0.0.1
adserver_revproxy_port = 80

[Task Parameters]
experiment_code_version = 1.0
numconds = 1
num_counters = 1
```

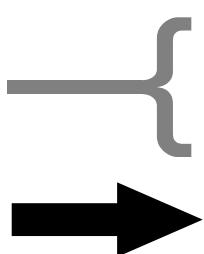


config.txt

```
exp/
  └── .psiturkconfig
  └── config.txt
  └── custom.py
  └── static/
  └── templates/
```

Server Parameters

- Correspond to default.conf
- Maximum time (min) participants may take to complete the experiment



```
require_master_workers = false

[Database Parameters]
database_url = mysql://paxton:psiturk@db:3306/participants
table_name = turkdemo

[Server Parameters]
host = 0.0.0.0
port = 22362
cutoff_time = 30
logfile = server.log
loglevel = 2
debug = true
threads = auto
adserver_revproxy_host = 127.0.0.1
adserver_revproxy_port = 80

[Task Parameters]
experiment_code_version = 1.0
numconds = 1
num_counters = 1
```



PSITURK

Server Parameters

- Correspond to `default.conf`
- Maximum time (min) participants may take to complete the experiment
- Correspond to `docker-compose.yml`

```
nginx:  
  container_name: my-experiment-nginx  
  image: nginx:latest  
  ports:  
    - 80:80  
  volumes:  
    - ./exp:/var/www/exp:ro  
    - ./default.conf:/etc/nginx/conf.d/default.conf  
  restart: unless-stopped
```

config.txt

```
require_master_workers = false
```

[Database Parameters]

```
database_url = mysql://paxton:psiturk@db:3306/participants  
table_name = turkdemo
```

[Server Parameters]

```
host = 0.0.0.0  
port = 22362  
cutoff_time = 30  
logfile = server.log  
loglevel = 2  
debug = true  
threads = auto  
adserver_revproxy_host = 127.0.0.1  
adserver_revproxy_port = 80
```

[Task Parameters]

```
experiment_code_version = 1.0  
num_conds = 1  
num_counters = 1
```

```
exp/  
  .psiturkconfig  
  config.txt  
  custom.py  
  static/  
  templates/
```



templates/

```
exp/
└── .psiturkconfig
└── config.txt
└── custom.py
└── static/
└── templates/
```



templates/

templates/

```
exp/
  └── .psiturkconfig
  └── config.txt
  └── custom.py
  └── static/
  └── templates/
```



templates/

- Study advertisement page

templates/
└── ad.html

```
exp/
└── .psiturkconfig
    config.txt
    custom.py
    static/
    templates/
```



templates/

- Study advertisement page
- Shown after completing experiment

```
templates/
└── ad.html
└── complete.html
```

```
exp/
└── .psiturkconfig
    config.txt
    custom.py
    static/
    templates/
```



templates/

- Study advertisement page
- Shown after completing experiment
- Fill in with IRB-approved consent form

```
templates/
└── ad.html
└── complete.html
└── consent.html
```

```
exp/
└── .psiturkconfig
    config.txt
    custom.py
    static/
    templates/
```



templates/

- Study advertisement page
- Shown after completing experiment
- Fill in with IRB-approved consent form
- Redirects to ad.html

```
templates/
  └── ad.html
  └── complete.html
  └── consent.html
  └── default.html
```

```
exp/
  └── .psiturkconfig
      config.txt
      custom.py
      static/
      templates/
```



templates/

- Study advertisement page
- Shown after completing experiment
- Fill in with IRB-approved consent form
- Redirects to ad.html
- Shown when an error occurs

```
templates/
  └── ad.html
  └── complete.html
  └── consent.html
  └── default.html
  └── error.html
```

```
exp/
  └── .psiturkconfig
      config.txt
      custom.py
      static/
      templates/
```



templates/

- Study advertisement page
- Shown after completing experiment
- Fill in with IRB-approved consent form
- Redirects to ad.html
- Shown when an error occurs
- The main experiment

```
templates/
  └── ad.html
  └── complete.html
  └── consent.html
  └── default.html
  └── error.html
  └── exp.html
```

```
exp/
  └── .psiturkconfig
  └── config.txt
  └── custom.py
  └── static/
  └── templates/
```



PSITURK

templates/

```
●●●

<!doctype html>
<html>
  <head>
    <title>Example experiment</title>

    <link rel="stylesheet" href="/static/css/bootstrap.min.css" type="text/css" />
    <link rel="stylesheet" href="/static/css/style.css" type="text/css" />
    <link rel="stylesheet" href="/static/css/jsppsych.css" type="text/css">

    <script src="/static/lib/jquery-min.js" type="text/javascript"></script>
    <script src="/static/lib/underscore-min.js" type="text/javascript"></script>
    <script src="/static/lib/backbone-min.js" type="text/javascript"></script>

    <script src="/static/js/jsppsych.js" type="text/javascript"></script>
    <script src="/static/js/jsppsych-fullscreen.js" type="text/javascript"></script>
    <script src="/static/js/jsppsych-html-button-response.js" type="text/javascript"></script>
    <script src="/static/js/jsppsych-html-keyboard-response.js" type="text/javascript"></script>
    <script src="/static/js/jsppsych-image-keyboard-response.js" type="text/javascript"></script>

    <script type="text/javascript">
      var uniqueId = "{{ uniqueId }}";
      var adServerLoc = "{{ adServerLoc }}";
      var mode = "{{ mode }}";
    </script>

    <script src="/static/js/utils.js" type="text/javascript"></script>
    <script src="/static/js/psiturk.js" type="text/javascript"></script>

    <script src="/static/js/experiment.js" type="text/javascript"></script>
  </head>
  <body>
    <noscript>
      <h1>Warning: Javascript seems to be disabled</h1>
      <p>This website requires that Javascript be enabled on your browser.</p>
      <p>Instructions for enabling Javascript in your browser can be found
        <a href="http://support.google.com/bin/answer.py?hl=en&answer=23852">here</a>
      </p>
    </noscript>
  </body>
</html>
```

```
templates/
  └── ad.html
  └── complete.html
  └── consent.html
  └── default.html
  └── error.html
  └── exp.html
```



templates/

```
templates/
└── ad.html
└── complete.html
└── consent.html
└── default.html
└── error.html
└── exp.html
```



```
<!doctype html>
<html>
  <head>
    <title>Example experiment</title>

    <link rel="stylesheet" href="/static/css/bootstrap.min.css" type="text/css" />
    <link rel="stylesheet" href="/static/css/style.css" type="text/css" />
    <link rel="stylesheet" href="/static/css/jsppsych.css" type="text/css">

    <script src="/static/lib/jquery-min.js" type="text/javascript"></script>
    <script src="/static/lib/underscore-min.js" type="text/javascript"></script>
    <script src="/static/lib/backbone-min.js" type="text/javascript"></script>

    <script src="/static/is/ispsvch.js" type="text/javascript"></script>
```



templates/

```
<title>Example experiment</title>

<link rel="stylesheet" href="/static/css/bootstrap.min.css" type="text/css" />
<link rel="stylesheet" href="/static/css/style.css" type="text/css" />
<link rel="stylesheet" href="/static/css/jspysch.css" type="text/css">

<script src="/static/lib/jquery-min.js" type="text/javascript"></script>
<script src="/static/lib/underscore-min.js" type="text/javascript"></script>
<script src="/static/lib/backbone-min.js" type="text/javascript"></script>

<script src="/static/js/jspysch.js" type="text/javascript"></script>
<script src="/static/js/jspysch-fullscreen.js" type="text/javascript"></script>
<script src="/static/js/jspysch-html-button-response.js" type="text/javascript"></script>
<script src="/static/js/jspysch-html-keyboard-response.js" type="text/javascript"></script>
<script src="/static/js/jspysch-image-keyboard-response.js" type="text/javascript"></script>

<script type="text/javascript">
    var uniqueId = "{{ uniqueId }}";
    var adServerLoc = "{{ adServerLoc }}";
    var mode = "{{ mode }}";
</script>
```

```
templates/
└── ad.html
└── complete.html
└── consent.html
└── default.html
└── error.html
└── exp.html
```



templates/

```
<script src="/static/js/jspysch-allscreen.js" type="text/javascript"></script>
<script src="/static/js/jspysch-html-button-response.js" type="text/javascript"></script>
<script src="/static/js/jspysch-html-keyboard-response.js" type="text/javascript"></script>
<script src="/static/js/jspysch-image-keyboard-response.js" type="text/javascript"></script>
```

```
templates/
└── ad.html
└── complete.html
└── consent.html
└── default.html
└── error.html
└── exp.html
```

```
<script type="text/javascript">
```

```
    var uniqueId = "{{ uniqueId }}";
    var adServerLoc = "{{ adServerLoc }}";
    var mode = "{{ mode }}";
</script>
```

```
<script src="/static/js/utils.js" type="text/javascript"></script>
```

```
<script src="/static/js/psiturk.js" type="text/javascript"></script>
```

```
<script src="/static/js/experiment.js" type="text/javascript"></script>
```

```
</head>
```

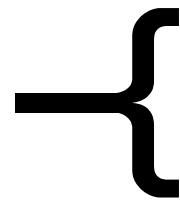
```
<body>
```

```
<noscript>
```

```
    <h1>Warning: Javascript seems to be disabled</h1>
```

```
    <p>This website requires that Javascript be enabled on your browser.</p>
```

```
    <p>Instructions for enabling Javascript in your browser can be found
```



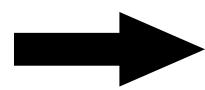
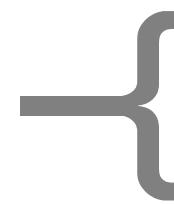
```
<script type="text/javascript">
    var uniqueId = "{{ uniqueId }}";
    var adServerLoc = "{{ adServerLoc }}";
    var mode = "{{ mode }}";
</script>

<script src="/static/js/utils.js" type="text/javascript"></script>
<script src="/static/js/psiturk.js" type="text/javascript"></script>

<script src="/static/js/experiment.js" type="text/javascript"></script>
</head>
<body>
    <noscript>
        <h1>Warning: Javascript seems to be disabled</h1>
        <p>This website requires that Javascript be enabled on your browser.</p>
        <p>Instructions for enabling Javascript in your browser can be found
```

- Required block, but fields provided by the psiTurk server

```
templates/
    └── ad.html
    ├── complete.html
    ├── consent.html
    ├── default.html
    ├── error.html
    └── exp.html
```



templates/

```
<script src="/static/js/jspysch-allscreen.js" type="text/javascript"></script>
<script src="/static/js/jspysch-html-button-response.js" type="text/javascript"></script>
<script src="/static/js/jspysch-html-keyboard-response.js" type="text/javascript"></script>
<script src="/static/js/jspysch-image-keyboard-response.js" type="text/javascript"></script>
```

```
templates/
└── ad.html
└── complete.html
└── consent.html
└── default.html
└── error.html
└── exp.html
```

```
<script type="text/javascript">
    var uniqueId = "{{ uniqueId }}";
    var adServerLoc = "{{ adServerLoc }}";
    var mode = "{{ mode }}";
</script>

<script src="/static/js/utils.js" type="text/javascript"></script>
<script src="/static/js/psiturk.js" type="text/javascript"></script>

<script src="/static/js/experiment.js" type="text/javascript"></script>
</head>
<body>
    <noscript>
        <h1>Warning: Javascript seems to be disabled</h1>
        <p>This website requires that Javascript be enabled on your browser.</p>
        <p>Instructions for enabling Javascript in your browser can be found
```

- Required block, but fields provided by the psiTurk server
- Generated dynamically at runtime



templates/

```
<script src="/static/js/psiturk.js" type="text/javascript"></script>
<script src="/static/js/psiturk.js" type="text/javascript"></script>

<script src="/static/js/experiment.js" type="text/javascript"></script>
</head>
<body>
<noscript>
    <h1>Warning: Javascript seems to be disabled</h1>
    <p>This website requires that Javascript be enabled on your browser.</p>
    <p>Instructions for enabling Javascript in your browser can be found
        <a href="http://support.google.com/bin/answer.py?hl=en&answer=23852">here</a>
    </p>
</noscript>
</body>
</html>
```

```
templates/
└── ad.html
└── complete.html
└── consent.html
└── default.html
└── error.html
└── exp.html
```



templates/

```
<script src="/static/js/jsPsych.js" type="text/javascript"></script>
<script src="/static/js/psiturk.js" type="text/javascript"></script>

<script src="/static/js/experiment.js" type="text/javascript"></script>
</head>
<body>
<noscript>
    <h1>Warning: Javascript seems to be disabled</h1>
    <p>This website requires that Javascript be enabled on your browser.</p>
    <p>Instructions for enabling Javascript in your browser can be found
        <a href="http://support.google.com/bin/answer.py?hl=en&answer=23852">here</a>
    </p>
</noscript>
</body>
</html>
```

```
templates/
└── ad.html
└── complete.html
└── consent.html
└── default.html
└── error.html
└── exp.html
```

- HTML body is empty, content is rendered by jsPsych



templates/

```
templates/
└── ad.html
└── complete.html
└── consent.html
└── default.html
└── error.html
└── exp.html
```



templates/

```
exp/
└── .psiturkconfig
└── config.txt
└── custom.py
└── static/
└── templates/
```



static/

```
exp/
└── .psiturkconfig
└── config.txt
└── custom.py
└── static/
└── templates/
```



static/

static/

```
exp/
└── .psiturkconfig
└── config.txt
└── custom.py
└── static/
└── templates/
```



static/

- Stylesheets for experiment pages

static/
└── css/

```
exp/  
  └── .psiturkconfig  
  └── config.txt  
  └── custom.py  
  └── static/  
  └── templates/
```



static/

- Stylesheets for experiment pages
- Font libraries

static/
└── css/
└── fonts/

```
exp/  
  └── .psiturkconfig  
  └── config.txt  
  └── custom.py  
  └── static/  
  └── templates/
```



static/

- Stylesheets for experiment pages
- Font libraries
- Experiment stimuli

static/
└── css/
└── fonts/
└── images/

```
exp/  
  └── .psiturkconfig  
  └── config.txt  
  └── custom.py  
  └── static/  
  └── templates/
```



static/

- Stylesheets for experiment pages
- Font libraries
- Experiment stimuli
- External JavaScript libraries

```
static/
  └── css/
  └── fonts/
  └── images/
  └── lib/
```

```
exp/
  └── .psiturkconfig
  └── config.txt
  └── custom.py
  └── static/
  └── templates/
```



static/

- Stylesheets for experiment pages
- Font libraries
- Experiment stimuli
- External JavaScript libraries
- Main experiment functionality

```
static/
  └── css/
  └── fonts/
  └── images/
  └── lib/
      └── js/
```

```
exp/
  └── .psiturkconfig
  └── config.txt
  └── custom.py
  └── static/
  └── templates/
```



static/

```
static/
  └── css/
      ├── bootstrap.min.css
      ├── style.css
      └── jsPsych.css
  └── fonts/
  └── images/
  └── lib/
      └── js/
```

```
exp/
  └── .psiturkconfig
  └── config.txt
  └── custom.py
  └── static/
  └── templates/
```



static/

```
static/
  └── css/
  └── fonts/
  └── images/
  └── lib/
  └── js/
```

```
exp/
  └── .psiturkconfig
  └── config.txt
  └── custom.py
  └── static/
  └── templates/
```



js/

```
static/  
  └── css/  
  └── fonts/  
  └── images/  
  └── lib/  
      └── js/
```



js/

js/

```
static/  
  └── css/  
  └── fonts/  
  └── images/  
  └── lib/  
      └── js/
```



js/

js/
└── experiment.js

```
static/
  └── css/
  └── fonts/
  └── images/
  └── lib/
    └── js/
```



js/

js/
└── experiment.js
└── utils.js

static/
├── css/
├── fonts/
├── images/
└── lib/
 └── js/



js/

js/
└── experiment.js
└── utils.js
└── jspsych.js
└── jspsych-html-button-response.js
└── jspsych-html-keyboard-response.js
└── jspsych-image-keyboard-response.js

static/
└── css/
└── fonts/
└── images/
└── lib/
└── js/



What is jsPsych?

```
static/
  └── css/
  └── fonts/
  └── images/
  └── lib/
    └── js/
```



What is jsPsych?

```
static/  
  -- css/  
  -- fonts/  
  -- images/  
  -- lib/  
  -- js/
```

- Javascript library for running behavioral experiments in the browser



What is jsPsych?

```
static/  
  -- css/  
  -- fonts/  
  -- images/  
  -- lib/  
  -- js/
```

- Javascript library for running behavioral experiments in the browser
- Main library with “plugins” for stages and tasks



What is jsPsych?

```
static/
  └── css/
  └── fonts/
  └── images/
  └── lib/
    └── js/
```

- Javascript library for running behavioral experiments in the browser
- Main library with “plugins” for stages and tasks
- Experiment constructed as a “timeline” of plugins



What is jsPsych?

```
static/
  └── css/
  └── fonts/
  └── images/
  └── lib/
    └── js/
```

- Javascript library for running behavioral experiments in the browser
- Main library with “plugins” for stages and tasks
- Experiment constructed as a “timeline” of plugins
- Tons of provided plugins & a template to build from



Why use jsPsych?

static/
 css/
 fonts/
 images/
 lib/
 js/



Why use jsPsych?

```
static/
  └── css/
  └── fonts/
  └── images/
  └── lib/
  └── js/
```

- Makes creating psiTurk experiment much easier



Why use jsPsych?

```
static/
  └── css/
  └── fonts/
  └── images/
  └── lib/
  └── js/
```

- Makes creating psiTurk experiment much easier
- Makes experiment code easier to understand



Why use jsPsych?

```
static/
  └── css/
  └── fonts/
  └── images/
  └── lib/
  └── js/
```

- Makes creating psiTurk experiment much easier
- Makes experiment code easier to understand
 - psiturk.js functions invisible, generated at runtime



experiment.js



experiment.js

```
js/
  └── experiment.js
  └── utils.js
  └── jspysch.js
  └── jspysch-html-button-...js
  └── jspysch-html-keyboard-...js
  └── jspysch-image-keyboard-...js
```



experiment.js

Go/No-go task

```
js/
  └── experiment.js
  └── utils.js
  └── jspsych.js
  └── jspsych-html-button-...js
  └── jspsych-html-keyboard-...js
  └── jspsych-image-keyboard-...js
```



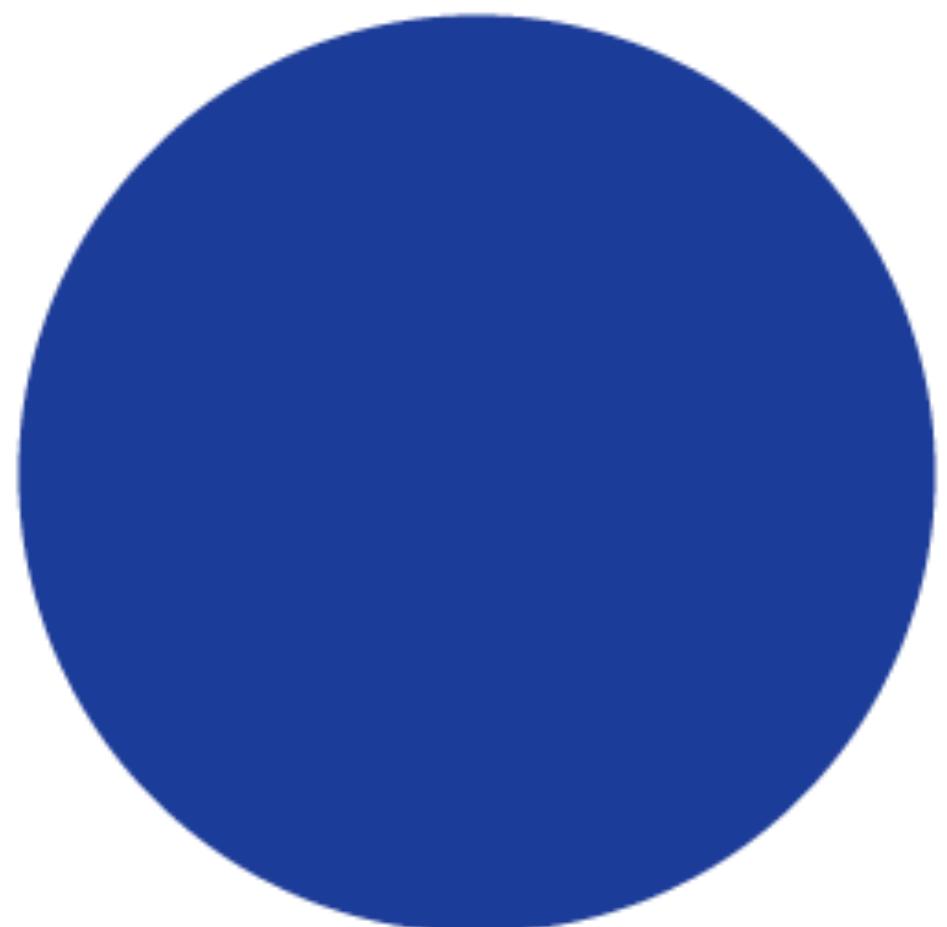
experiment.js

Go/No-go task

In this experiment, a circle will appear in the center of the screen.

If the circle is **blue**, press the letter "**f**" on the keyboard as fast as you can.

If the circle is **orange**, **do not press any key**.



Press the F key



Do not press a key

Press any key to begin.

```
js/
  └── experiment.js
  └── utils.js
  └── jspsych.js
  └── jspsych-html-button-...js
  └── jspsych-html-keyboard-...js
  └── jspsych-image-keyboard-...js
```

Demo

