# edX Load Tests

## Introduction And Current Status

Troy Sankey

September 5, 2017

# Introduction

- Tests live in the edx/edx-load-tests repo.

# Introduction

- Tests live in the edx/edx-load-tests repo.
- Uses the Locust load testing tool.

# Why?

- Performance-related improvements.

# Why?

- Performance-related improvements.
- Performance-sensitive code changes.
- Major version changes.

# locustfile.py

```python
class ChatTasks(TaskSet)
    @task(2)
    def open_chat():
        print("opening a chat")
    @task(3)
    def send_message():
        print("sending a message")


class ServiceTasks(TaskSet):
    tasks = {
        ChatTasks: 1,
        SearchTasks: 2,
    }

class ServiceUser(HttpLocust):
    task_set = ServiceTasks
    min_wait = 1000
    max_wait = 5000
```
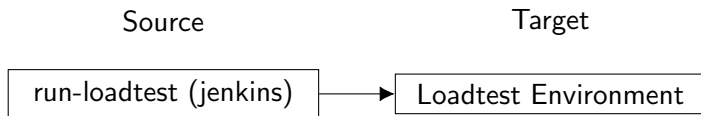
## locustfile.py

```python
class ChatTasks(TaskSet)
    @task(2)
    def open_chat():
        print("opening a chat")
    @task(3)
    def send_message():
        print("sending a message")


class ServiceTasks(TaskSet):
    tasks = {
        ChatTasks: 1,
        SearchTasks: 2,
    }

class ServiceUser(HttpLocust):
    task_set = ServiceTasks
    min_wait = 1000
    max_wait = 5000
```

# locustfile.py

```python
class ChatTasks(TaskSet)
    @task(2)
    def open_chat():
        print("opening a chat")
    @task(3)
    def send_message():
        print("sending a message")


class ServiceTasks(TaskSet):
    tasks = {
        ChatTasks: 1,
        SearchTasks: 2,
    }

class ServiceUser(HttpLocust):
    task_set = ServiceTasks
    min_wait = 1000
    max_wait = 5000
```

# User Experience

- No browser, no javascript, no good measure of user experience.
- Focus on the system response, not the users.

# Overview Of Running Load Tests

Target

Loadtest Environment
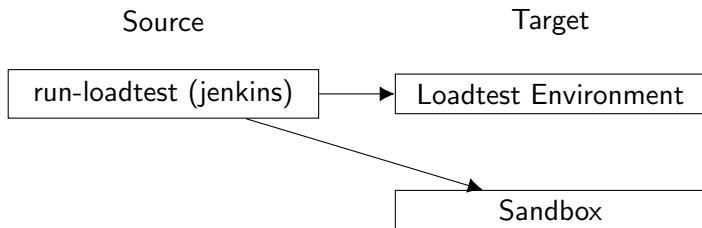
# Overview Of Running Load Tests

Source                                Target

| run-loadtest (jenkins) | → | Loadtest Environment |

# Overview Of Running Load Tests

Source                          Target

| run-loadtest (jenkins) | → | Loadtest Environment |

| Sandbox |

# Overview Of Running Load Tests
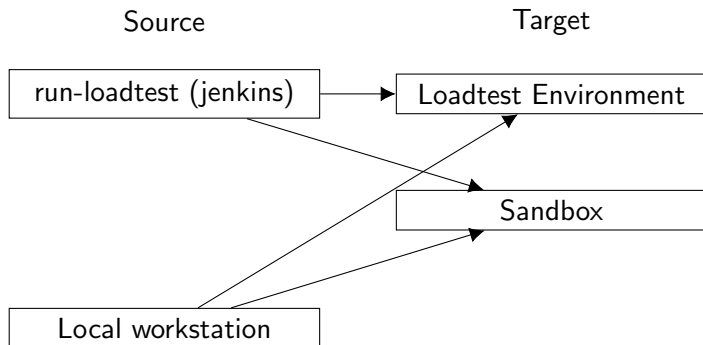
# Overview Of Running Load Tests
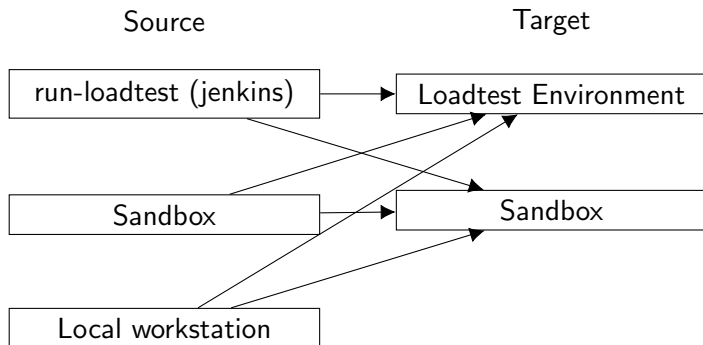
# Overview Of Running Load Tests
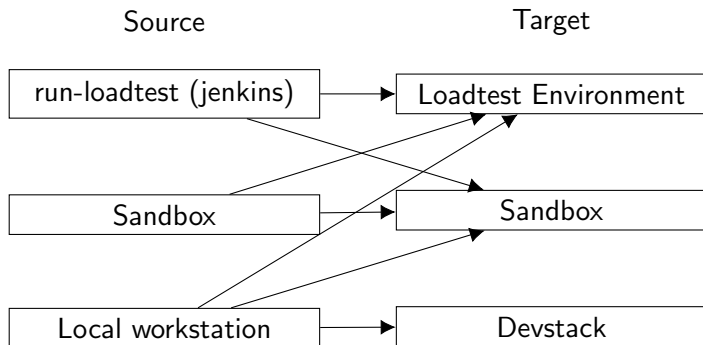
# Overview Of Running Load Tests
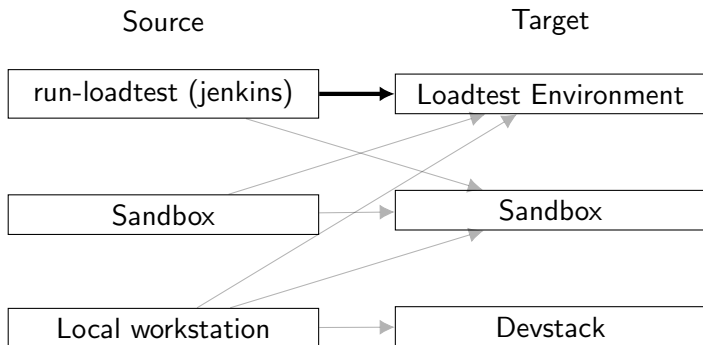
# Settings

- Each load test is configurable via a YAML file.

# Settings

- Each load test is configurable via a YAML file.
- Use `helpers.settings`.

# Settings

- Each load test is configurable via a YAML file.
- Use `helpers.settings`.
- The Jenkins job (run-loadtest) will use settings specified by a job parameter.

# Evaluating Load Test Runs

- Locust provides response time breakdown.

# Evaluating Load Test Runs

- Locust provides response time breakdown.
- Click on the summary artifact.

# Evaluating Load Test Runs

- ▶ Locust provides response time breakdown.
- ▶ Click on the summary artifact.



- ▶ NewRelic provides deeper application insight.

# Evaluating Load Test Runs

- ▶ Locust provides response time breakdown.
- ▶ Click on the summary artifact.



- ▶ NewRelic provides deeper application insight.
- ▶ Custom metrics in edx-platform: make use of the custom metrics middleware for peering into application behavior.

# Record Your Findings

- Create a wiki page before testing.

# Record Your Findings

- Create a wiki page before testing.
- Caution: Beware of NewRelic data atrophie. NR gets hungry and eats your old dots.

# Record Your Findings

- Create a wiki page before testing.
- Caution: Beware of NewRelic data atrophie. NR gets hungry and eats your old dots.
- Continuous deployment $\Rightarrow$ continuous deletion of system metrics!

# Distributed load testing

- Just run multiple load tests simultaneously in Jenkins.

# Distributed load testing

- Just run multiple load tests simultaneously in Jenkins.
- You may need to prime N workers first.

# Distributed load testing

- Just run multiple load tests simultaneously in Jenkins.
- You may need to prime N workers first.
- If you want/need real distributed load testing, demand it!

# How to maintain load tests

- Task ratios will rot.

# How to maintain load tests

- Task ratios will rot.
- Tasks themselves will rot.

# How to maintain load tests

- Task ratios will rot.
- Tasks themselves will rot.
- New endpoints will need to be added as new tasks.

# For fun: Real browser load testing

- `pip install realbrowserlocusts`, then subclass PhantomJSLocust in your locustfile.
- Could make certain tests more realistic (AJAX calls actually happen).
- Potentially far more resource intensive on the client side