How to Write Deployment-friendly Applications

Hynek Schlawack

VIEWS.PY

```
from pyramid.response import Response
from pyramid.view import view_config

@view_config(route_name="hello")
def hello_world(request):
    return Response("Hello World!")
```

APP_MAKER.PY

```
from pyramid.config import Configurator
from .views import hello_world

def make_app():
    config = Configurator()
    config.add_route("hello", "/hello")
    config.scan()

    return config.make_wsgi_app()
```

APP_MAKER.PY

```
from pyramid.config import Configurator
from .views import hello_world


def make_app():
    config = Configurator()
    config.add_route("hello", "/hello")
    config.scan()

    return config.make_wsgi_app()
```

```
$ gunicorn \
    --access-logfile - \
    "sample.app_maker:make_app()"
```

```
$ gunicorn \
    --access-logfile - \
    "sample.app_maker:make_app()"
[2018-04-05 16:28:06 +0200] [54343] [INFO] Starting gunicorn 19.7.1
[2018-04-05 16:28:06 +0200] [54343] [INFO] Listening at: http://127.0.0.1:8000 (54343)
[2018-04-05 16:28:06 +0200] [54343] [INFO] Using worker: sync
[2018-04-05 16:28:06 +0200] [54346] [INFO] Booting worker with pid: 54346
```

```
$ gunicorn \
    --access-logfile - \
    "sample.app_maker:make_app()"
[2018-04-05 16:28:06 +0200] [54343] [INFO] Starting gunicorn 19.7.1
[2018-04-05 16:28:06 +0200] [54343] [INFO] Listening at: http://127.0.0.1:8000 (54343)
[2018-04-05 16:28:06 +0200] [54343] [INFO] Using worker: sync
[2018-04-05 16:28:06 +0200] [54346] [INFO] Booting worker with pid: 54346
```

```
$ curl http://127.0.0.1:8000/hello
Hello World!
```

```
$ gunicorn \
    --access-logfile - \
    "sample.app_maker:make_app()"
[2018-04-05 16:28:06 +0200] [54343] [INFO] Starting gunicorn 19.7.1
[2018-04-05 16:28:06 +0200] [54343] [INFO] Listening at: http://127.0.0.1:8000 (54343)
[2018-04-05 16:28:06 +0200] [54343] [INFO] Using worker: sync
[2018-04-05 16:28:06 +0200] [54346] [INFO] Booting worker with pid: 54346
127.0.0.1 - - [05/Apr/2018:16:28:08 +0200] "GET /hello HTTP/1.1" 200 12 "-" "curl/7.54.0"

$ curl http://127.0.0.1:8000/hello
Hello World!
```

Zoom Out

RUN-APP.SH

```
#!/bin/bash

exec 2>&1 \
   gunicorn \
      --access-logfile - \
      "sample.app_maker:make_app()"
```

RUN-APP.SH

```
#!/bin/bash

exec 2>&1 \
   gunicorn \
      --access-logfile - \
      "sample.app_maker:make_app()"
```

RUN-APP.SH

```
#!/bin/bash

exec 2>&1 \
   gunicorn \
      --access-logfile - \
      "sample.app_maker:make_app()"
```

```
ExecStart=/app/run-app.sh
```

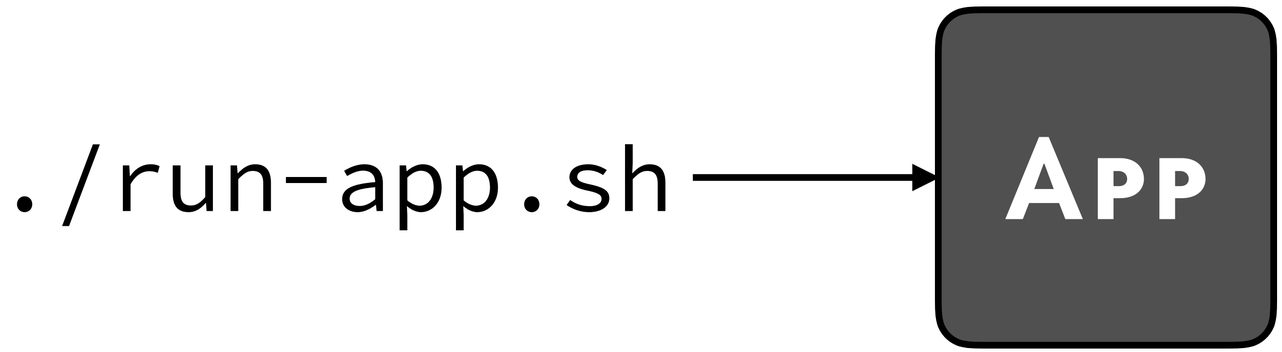
Dockerfile

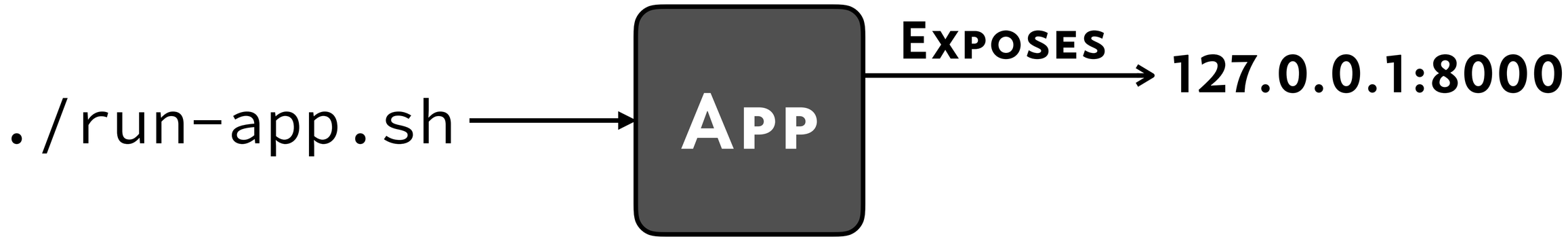
```
ENTRYPOINT ["/app/run-app.sh"]
```

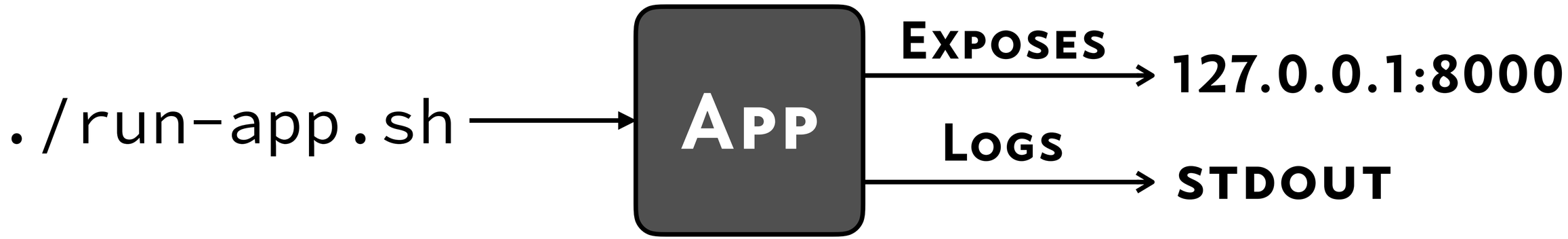
Procfile

```
web: ./run-app.sh
```


App


./run-app.sh
App


./run-app.sh
App
Exposes
127.0.0.1:8000


./run-app.sh
App
Exposes
127.0.0.1:8000
Logs
stdout

Configuration

What Varies?

What Varies?

Very Little

```
[server]
host=127.0.0.1
port=8000
log_level=INFO
log_format=human
```

Environment Variables

```
Environment=LOG_FORMAT=human
Environment=LOG_LEVEL=INFO
ExecStart=/app/run-app.sh
```

Dockerfile

```
ENV LOG_FORMAT=human
ENV LOG_LEVEL=INFO

ENTRYPOINT ["/app/run-app.sh"]
```

DIRENV

ENVCONSUL

DIRENV

ETCDENV

ENVCONSUL

DIRENV

ETCDENV

os.environ

DIRENV

ENVCONSUL

ETCDENV

ENVSUBST

os.environ

CONFD

DIRENV

ENVCONSUL

ETCDENV

CONSUL-TEMPLATE

ENVSUBST

os.environ

```
$ env HOST=0.0.0.0 PORT=8888 LOG_LEVEL=INFO \
    ./run-app.sh
[2018-04-09 15:59:29 +0200] [35323] [INFO] Starting gunicorn 19.7.1
[2018-04-09 15:59:29 +0200] [35323] [INFO] Listening at: http://0.0.0.0:8888
```

```
$ env HOST=0.0.0.0 PORT=8888 LOG_LEVEL=INFO \
    ./run-app.sh
[2018-04-09 15:59:29 +0200] [35323] [INFO] Starting gunicorn 19.7.1
[2018-04-09 15:59:29 +0200] [35323] [INFO] Listening at: http://0.0.0.0:8888
```

```
$ env HOST=0.0.0.0 PORT=8888 LOG_LEVEL=INFO \
    ./run-app.sh
[2018-04-09 15:59:29 +0200] [35323] [INFO] Starting gunicorn 19.7.1
[2018-04-09 15:59:29 +0200] [35323] [INFO] Listening at: http://0.0.0.0:8888
```

```
$ env HOST=0.0.0.0 PORT=8888 LOG_LEVEL=INFO \
    ./run-app.sh
[2018-04-09 15:59:29 +0200] [35323] [INFO] Starting gunicorn 19.7.1
[2018-04-09 15:59:29 +0200] [35323] [INFO] Listening at: http://0.0.0.0:8888
```

```
logging.basicConfig(
    level=getattr(
        logging,
        os.environ["LOG_LEVEL"],
    ),
    format="%(message)s",
    stream=sys.stdout,
)
```

CONFIG.PY

```
import environ

@environ.config(prefix="APP")
class AppConfig:
    @environ.config
    class Log:
        level = environ.var()
        format = environ.var()

    log = environ.group(Log)
```

CONFIG.PY

```
import environ

@environ.config(prefix="APP")
class AppConfig:
    @environ.config
    class Log:
        level = environ.var()
        format = environ.var()

    log = environ.group(Log)
```

APP_LOG_LEVEL → app_cfg.**log.level**

WSGI.PY

```
import environ

from .config import AppConfig
from .app_maker import make_app

app_cfg = environ.to_config(AppConfig)

application = make_app(app_cfg)
```

WSGI.PY

```
import environ

from .config import AppConfig
from .app_maker import make_app

app_cfg = environ.to_config(AppConfig)

application = make_app(app_cfg)
```

RUN-APP.SH

```
#!/bin/bash

exec 2>&1 \
   gunicorn \
      --access-logfile - \
      "sample.wsgi"
```

Don't Put Sensitive Data Into Env Variables

Google Cloud KMS

Microsoft Azure Key Vault

Docker Secrets

AWS Secrets Manager

HashiCorp Vault

Google Cloud KMS

Microsoft Azure Key Vault

Docker Secrets

AWS Secrets Manager

HashiCorp Vault

SECRETS.PY

SECRETS.PY

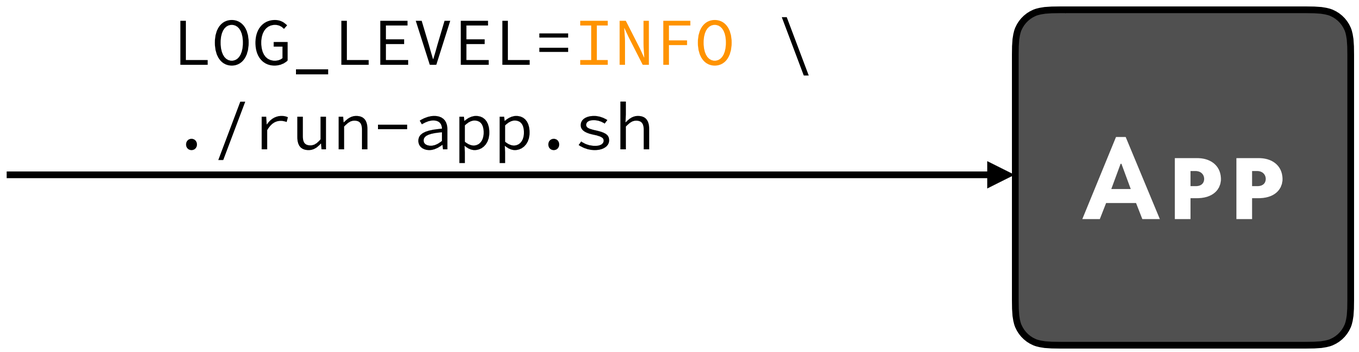
```python
class VaultSecrets:
    # ...
    def get_db_url(self):
        return self.vault.read(
            f"secret/{self.env}/app-name"
        )["data"]["db_url"])
```

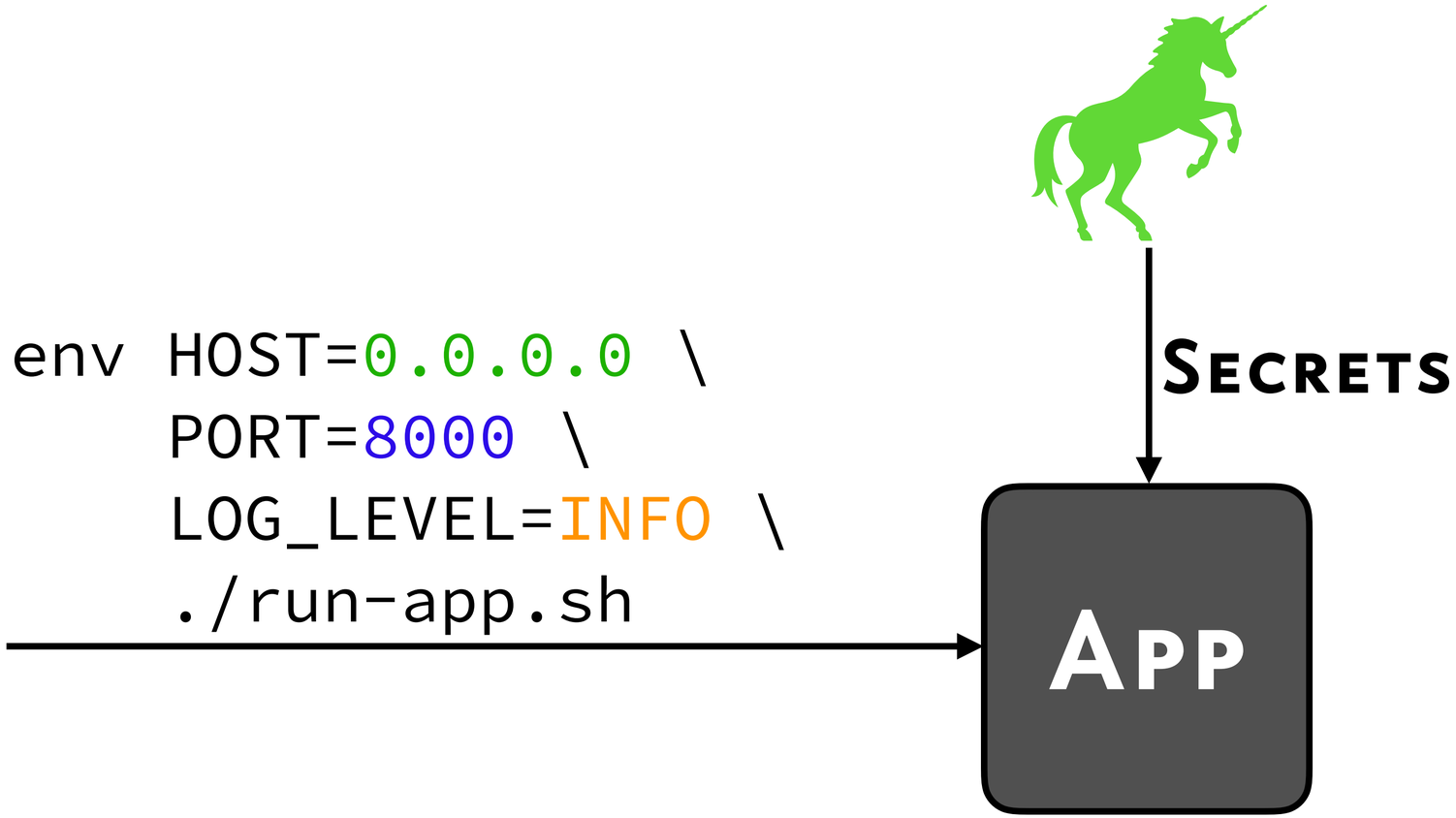
SECRETS.PY

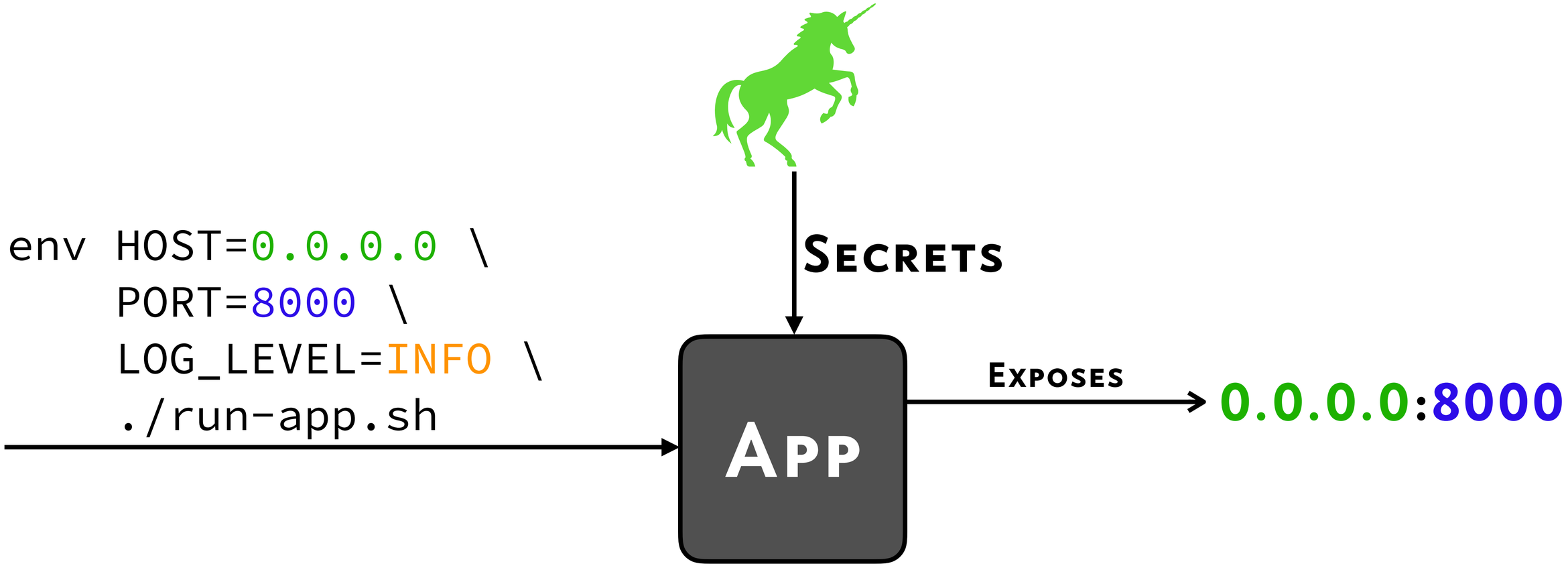
```
class FakeSecrets:
    def get_db_url(self):
        return (
            "postgresql://user@localhost/db"
        )
```

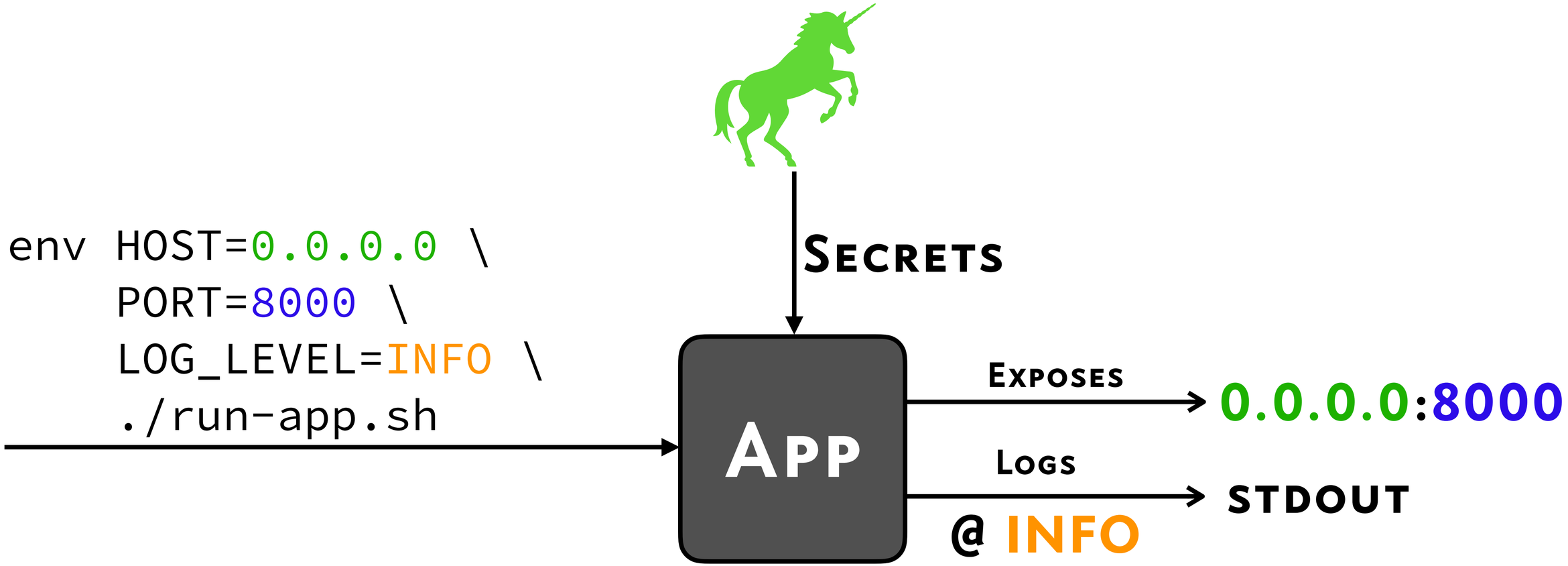

App

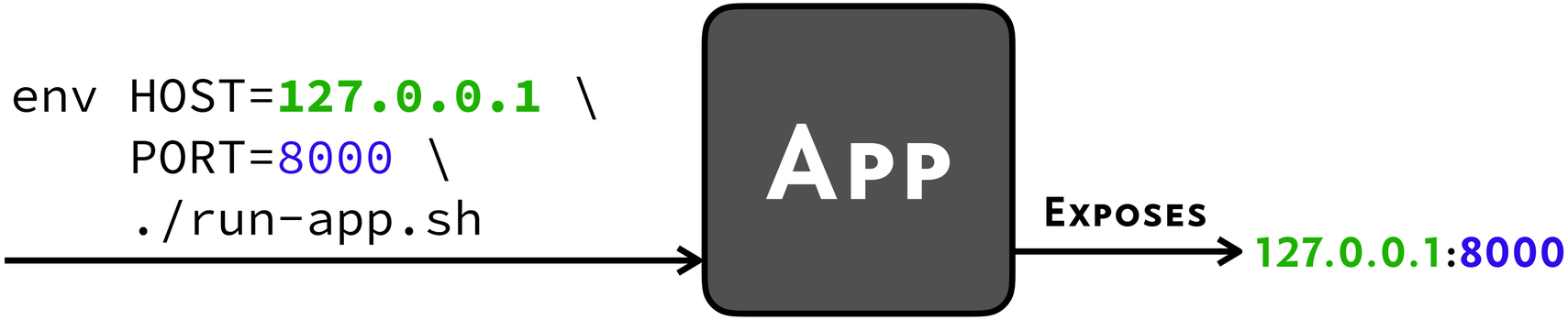
```
env HOST=0.0.0.0 \
    PORT=8000 \
    LOG_LEVEL=INFO \
    ./run-app.sh
```

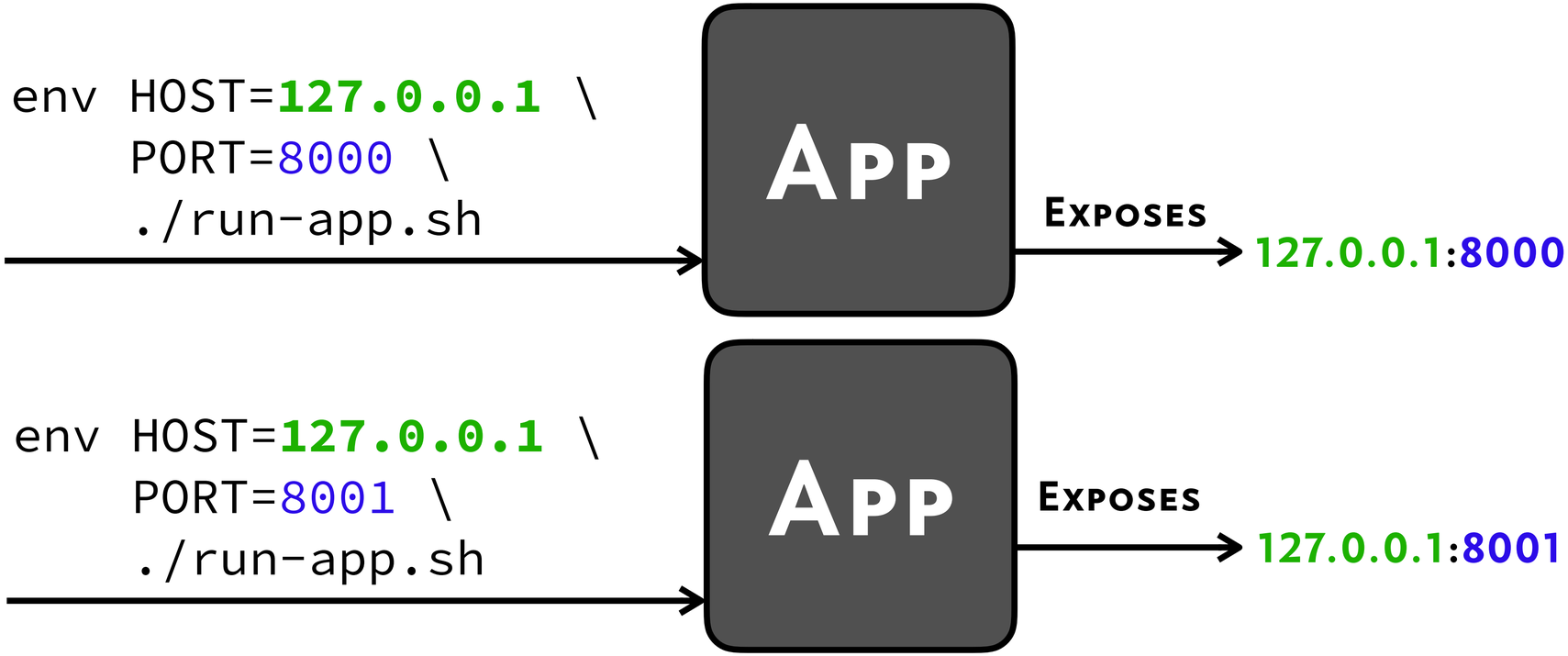


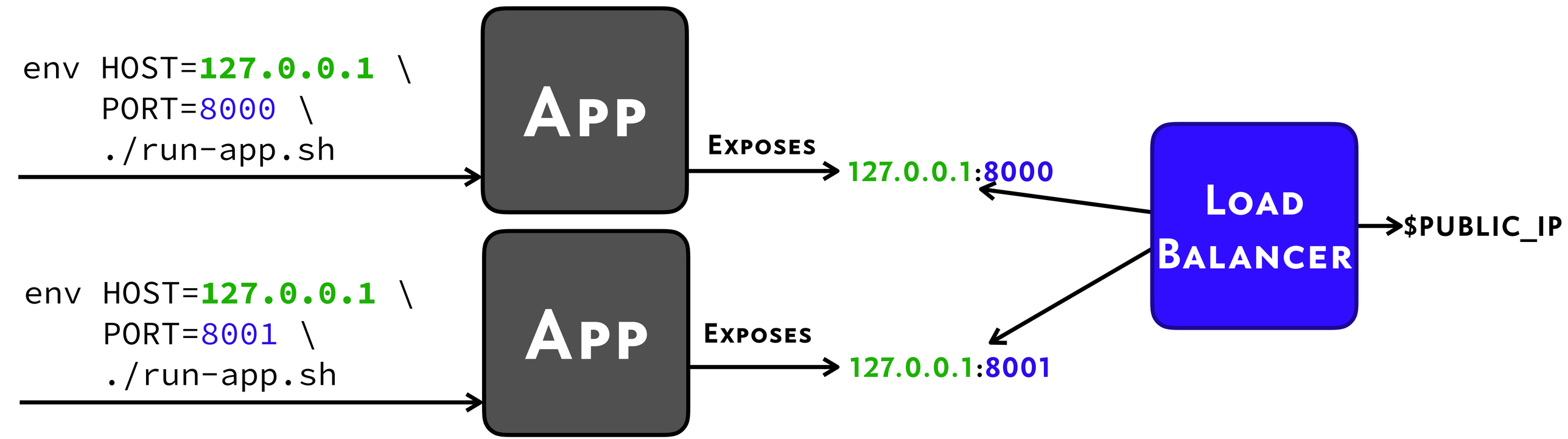

env HOST=0.0.0.0 \
PORT=8000 \
LOG_LEVEL=INFO \
./run-app.sh
Secrets
App

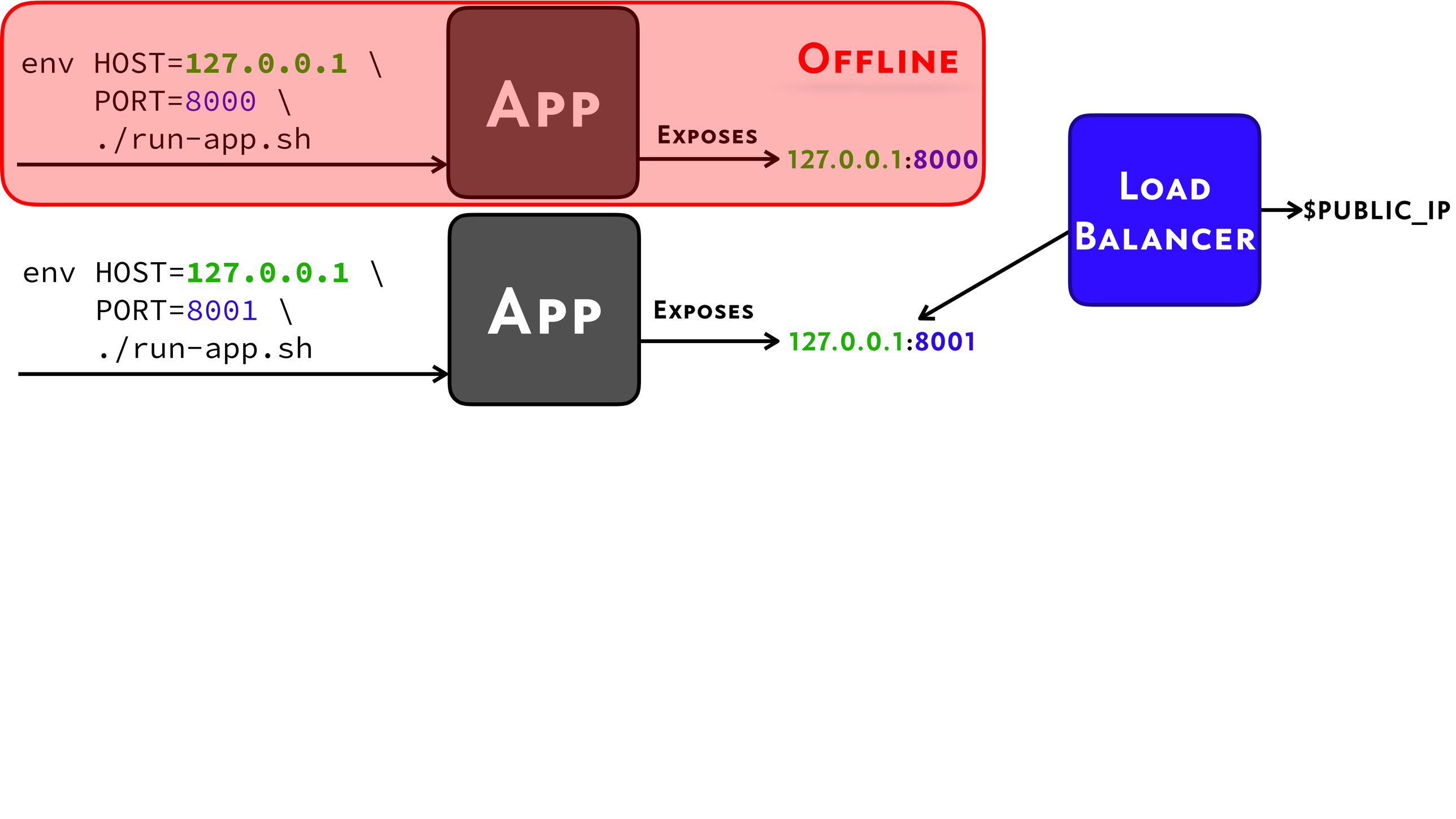

env HOST=0.0.0.0 \
PORT=8000 \
LOG_LEVEL=INFO \
./run-app.sh
Secrets
App
Exposes
0.0.0.0:8000


env HOST=0.0.0.0 \
PORT=8000 \
LOG_LEVEL=INFO \
./run-app.sh
Secrets
App
Exposes
0.0.0.0:8000
Logs
stdout
@ INFO

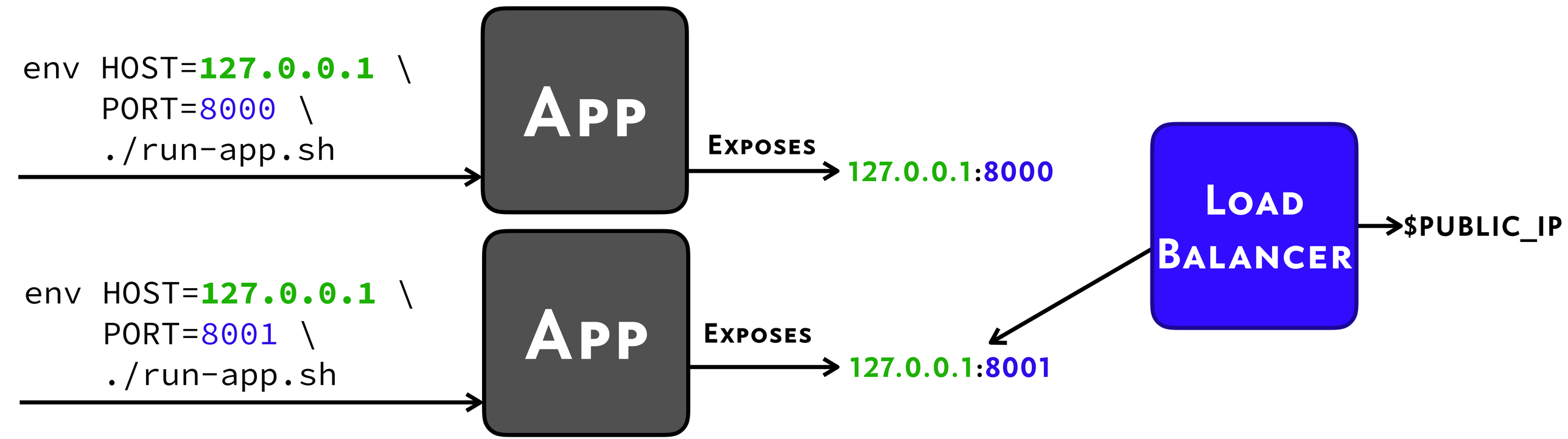

env HOST=127.0.0.1 \
PORT=8000 \
./run-app.sh
App
Exposes
127.0.0.1:8000

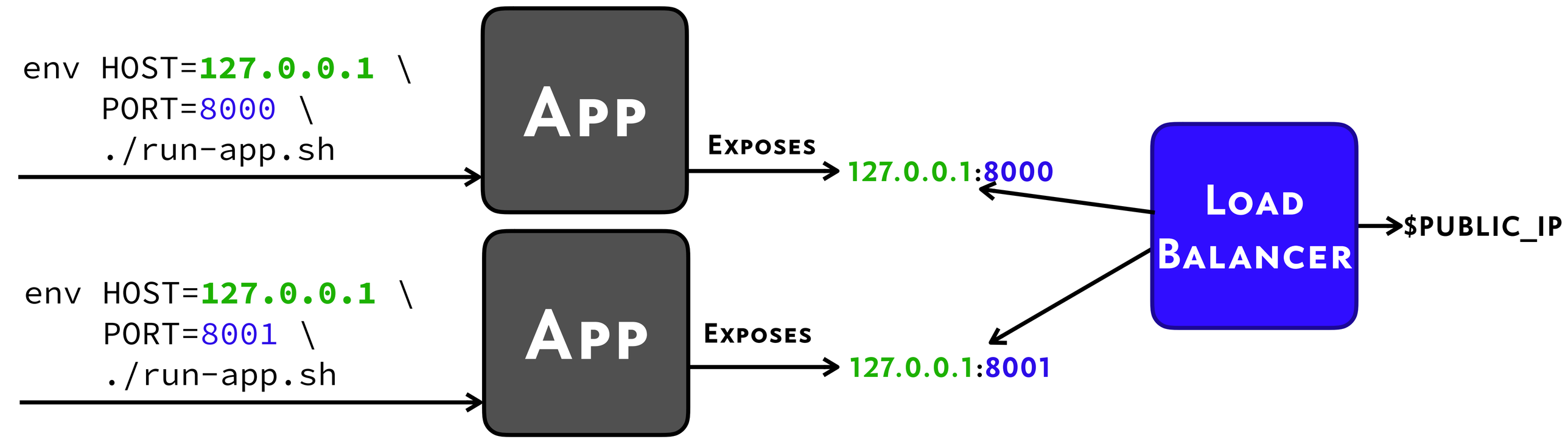

env HOST=127.0.0.1 \
PORT=8000 \
./run-app.sh
App
Exposes
127.0.0.1:8000
env HOST=127.0.0.1 \
PORT=8001 \
./run-app.sh
App
Exposes
127.0.0.1:8001

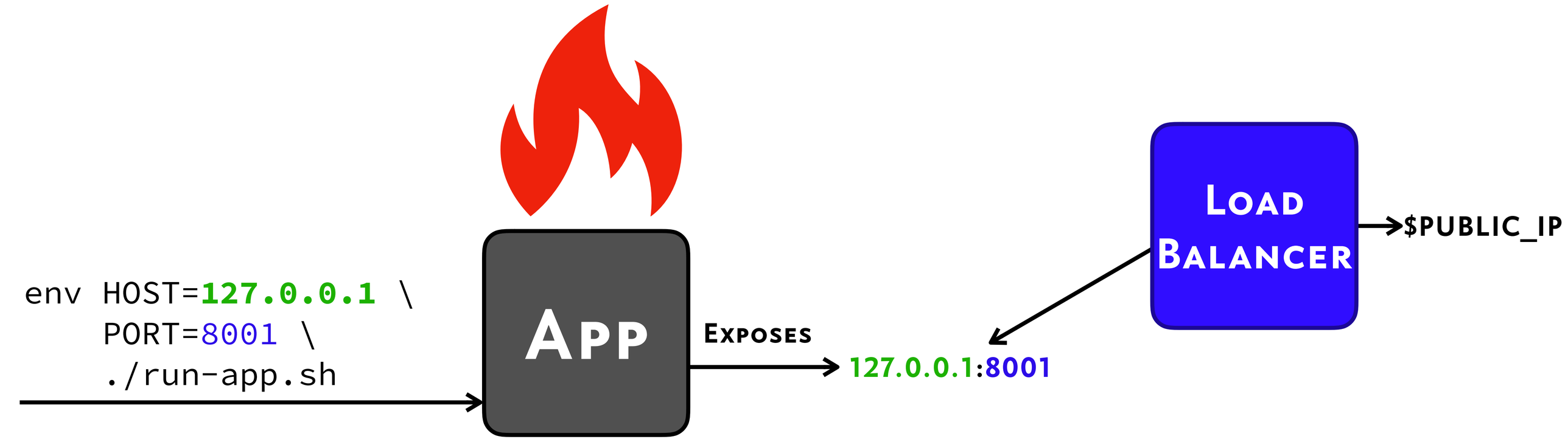

env HOST=127.0.0.1 \
PORT=8000 \
./run-app.sh
App
Exposes
127.0.0.1:8000
Load Balancer
$PUBLIC_IP
env HOST=127.0.0.1 \
PORT=8001 \
./run-app.sh
App
Exposes
127.0.0.1:8001


env HOST=127.0.0.1 \
PORT=8000 \
./run-app.sh
App
Offline
Exposes
127.0.0.1:8000
Load Balancer
$PUBLIC_IP
env HOST=127.0.0.1 \
PORT=8001 \
./run-app.sh
App
Exposes
127.0.0.1:8001

SIGTERM


env HOST=127.0.0.1 \
PORT=8000 \
./run-app.sh
App
Exposes
127.0.0.1:8000
env HOST=127.0.0.1 \
PORT=8001 \
./run-app.sh
App
Exposes
127.0.0.1:8001
Load Balancer
$PUBLIC_IP


env HOST=127.0.0.1 \
PORT=8000 \
./run-app.sh
App
Exposes
127.0.0.1:8000
Load Balancer
$PUBLIC_IP
env HOST=127.0.0.1 \
PORT=8001 \
./run-app.sh
App
Exposes
127.0.0.1:8001


env HOST=127.0.0.1 \
PORT=8001 \
./run-app.sh
App
Exposes
127.0.0.1:8001
Load Balancer
$PUBLIC_IP

Readiness

Readiness

- /healthz

Readiness

- /healthz
- /__heartbeat__

Readiness

- /healthz
- /__heartbeat__
- /-/ready

Readiness

- /healthz
- /__heartbeat__
- /-/ready
- /-/readiness

HAProxy

```
http-request deny if { path_beg /-/ }
```

Liveness

Liveness

- /-/healthy

Liveness

- /-/healthy
- /-/liveness

Liveness

- /-/healthy
- /-/liveness
- __lbheartbeat__

```
@view_config(
      route_name="ready",
      permission=NO_PERMISSION_REQUIRED,
)
def ready(request):
    return Response("yep")
```

```python
@view_config(
        route_name="ready",
        permission=NO_PERMISSION_REQUIRED,
)
def ready(request):
    return Response("yep")
```

```python
config.add_route("ready", "/-/ready")
```

Moar

Moar

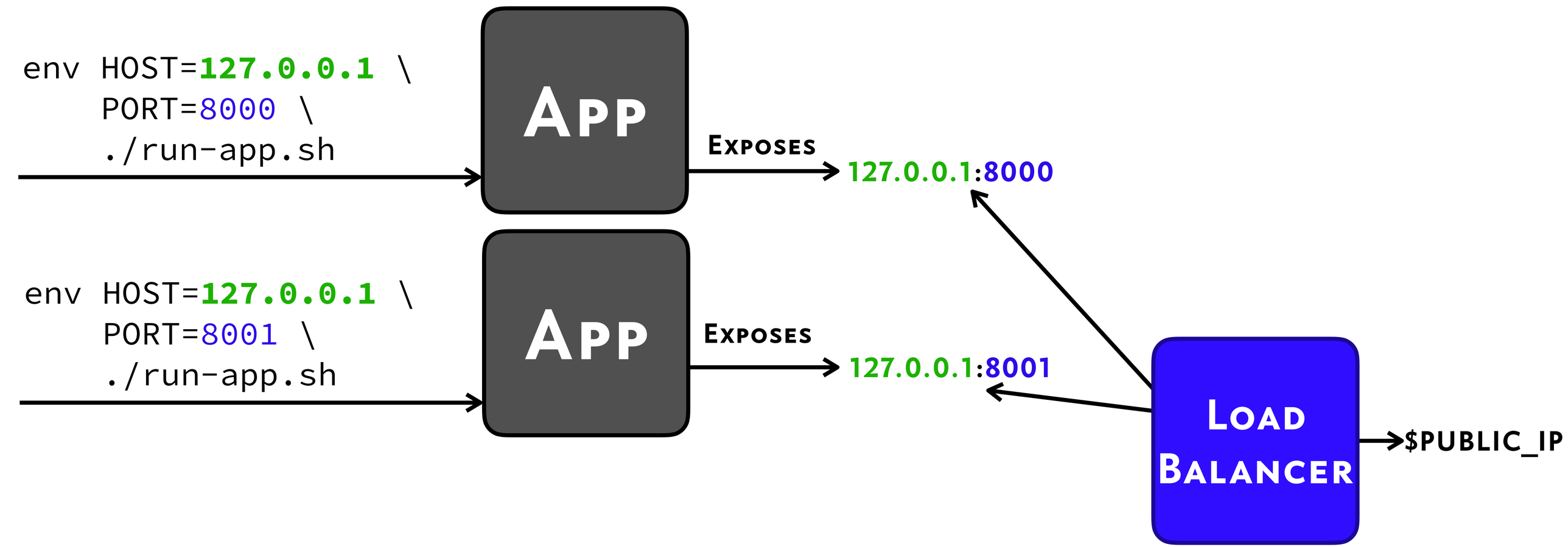
- `__version__`

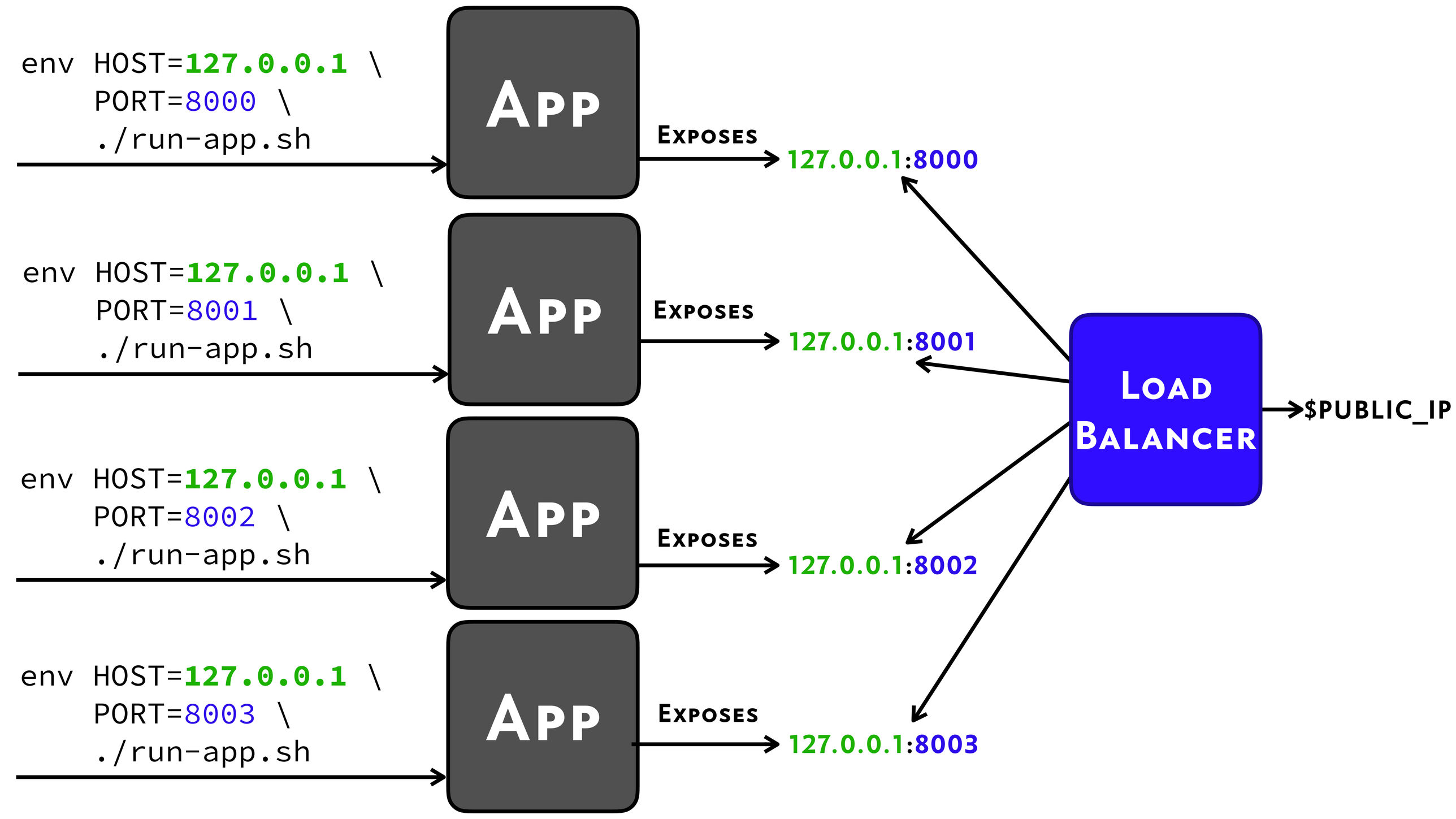
Moar

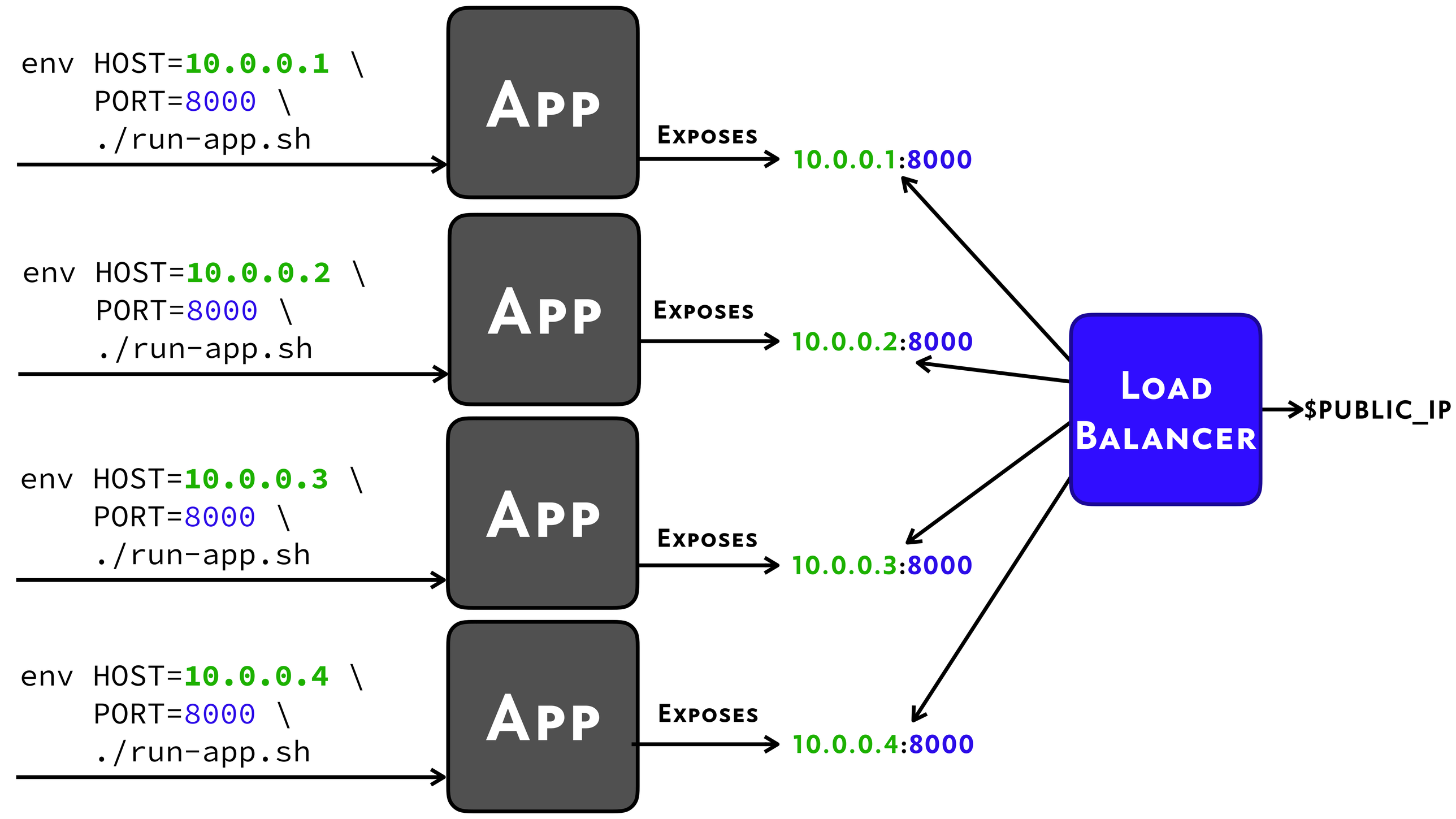
- __version__
- /-/metrics

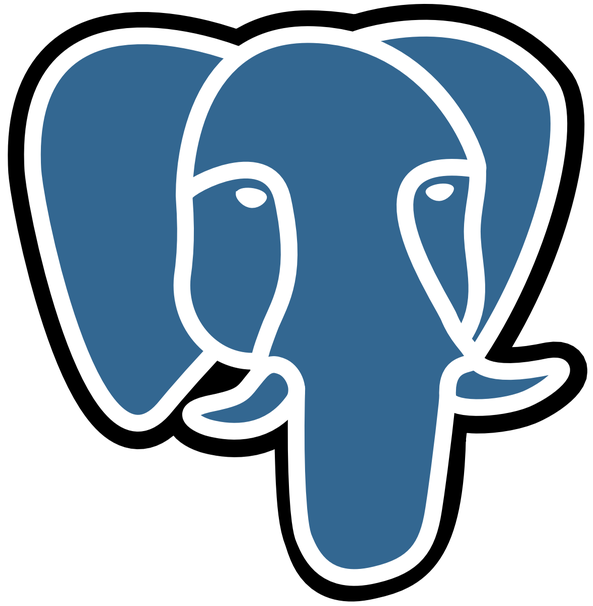
Moar

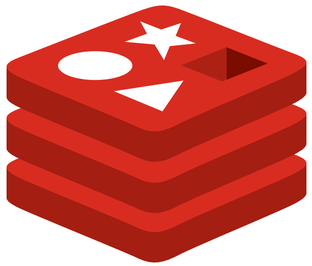
- __version__
- /-/metrics
- /-/log-level


env HOST=127.0.0.1 \
PORT=8000 \
./run-app.sh
App
Exposes
127.0.0.1:8000
env HOST=127.0.0.1 \
PORT=8001 \
./run-app.sh
App
Exposes
127.0.0.1:8001
Load Balancer
$PUBLIC_IP

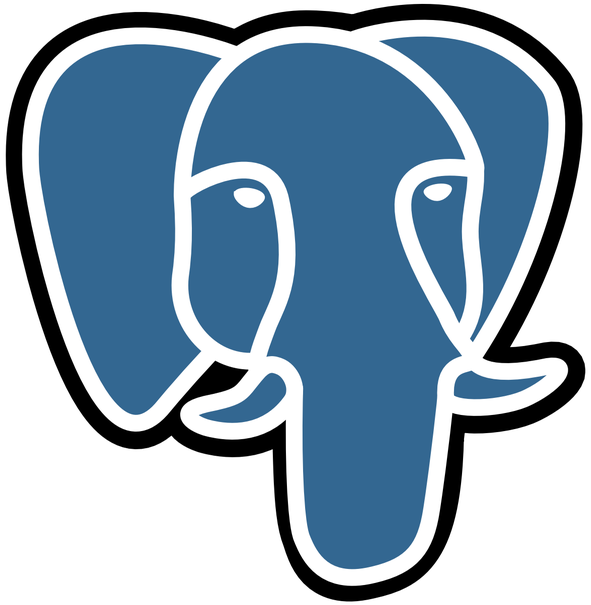

env HOST=127.0.0.1 \
PORT=8000 \
./run-app.sh
App
Exposes
127.0.0.1:8000
env HOST=127.0.0.1 \
PORT=8001 \
./run-app.sh
App
Exposes
127.0.0.1:8001
env HOST=127.0.0.1 \
PORT=8002 \
./run-app.sh
App
Exposes
127.0.0.1:8002
env HOST=127.0.0.1 \
PORT=8003 \
./run-app.sh
App
Exposes
127.0.0.1:8003
Load Balancer
$PUBLIC_IP

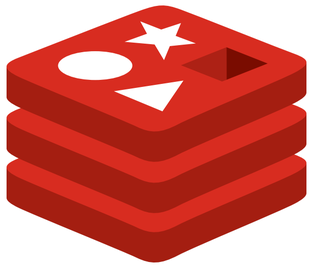

env HOST=10.0.0.1 \
PORT=8000 \
./run-app.sh
App
Exposes
10.0.0.1:8000
env HOST=10.0.0.2 \
PORT=8000 \
./run-app.sh
App
Exposes
10.0.0.2:8000
env HOST=10.0.0.3 \
PORT=8000 \
./run-app.sh
App
Exposes
10.0.0.3:8000
env HOST=10.0.0.4 \
PORT=8000 \
./run-app.sh
App
Exposes
10.0.0.4:8000
Load Balancer
$PUBLIC_IP



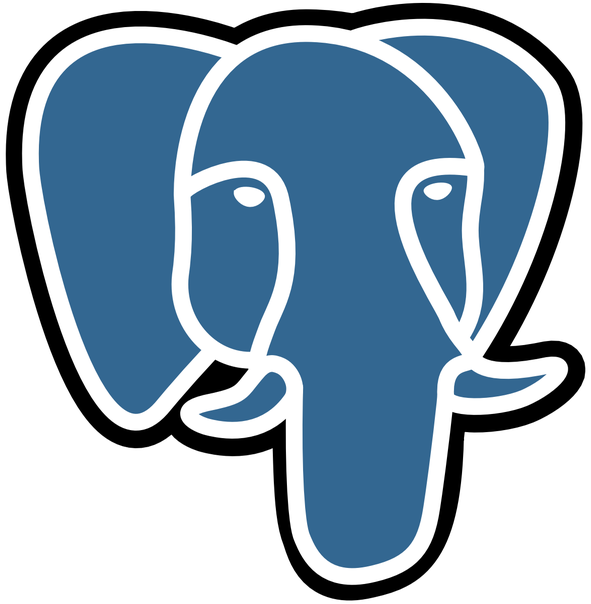


redis

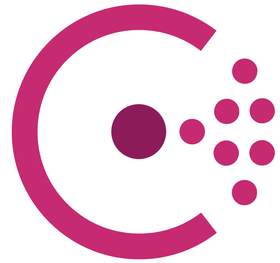




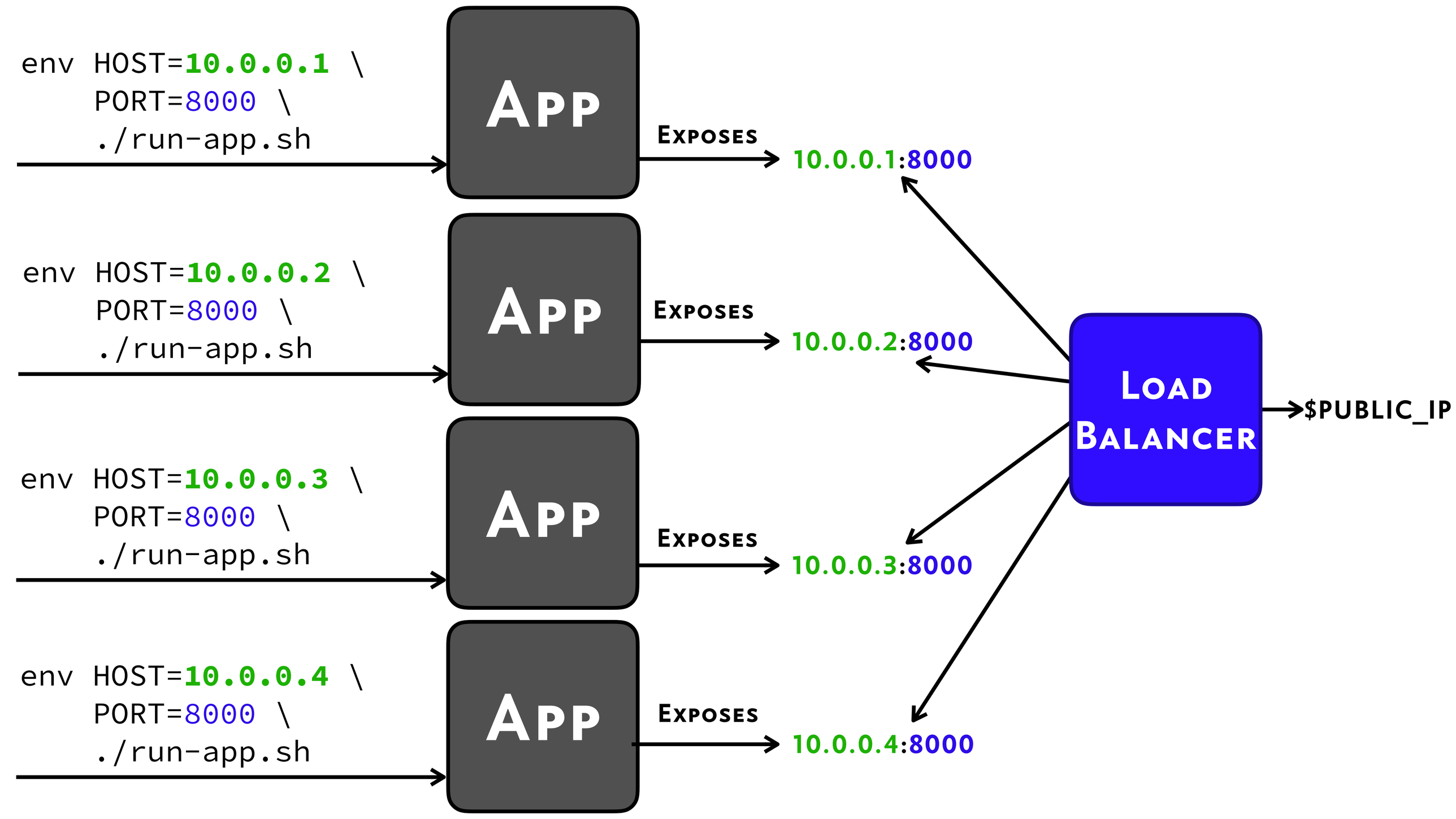
redis

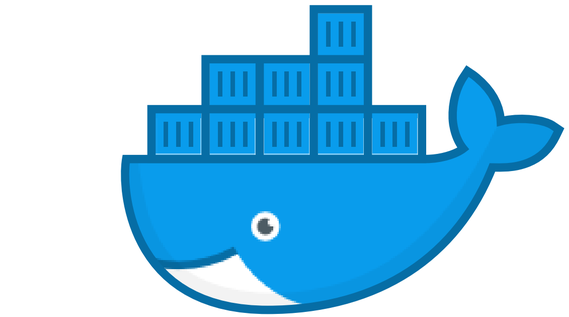


etcd

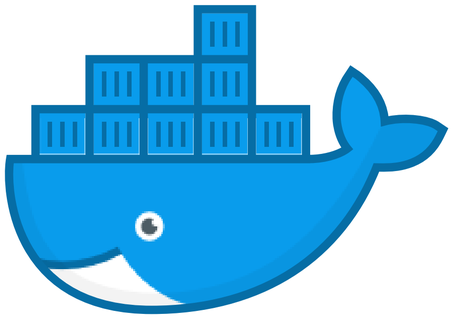


Consul

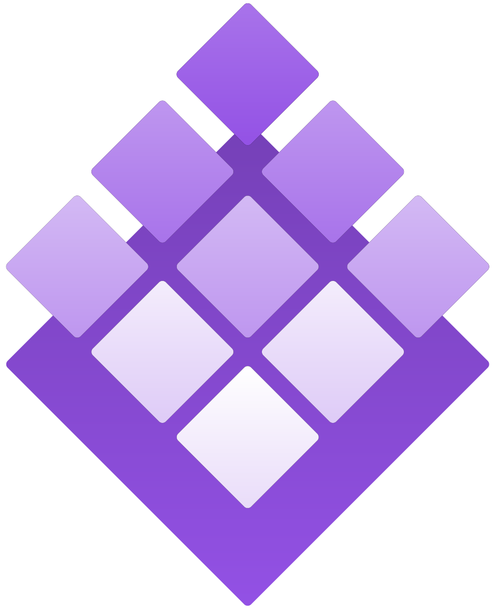

env HOST=10.0.0.1 \
PORT=8000 \
./run-app.sh
App
Exposes
10.0.0.1:8000
env HOST=10.0.0.2 \
PORT=8000 \
./run-app.sh
App
Exposes
10.0.0.2:8000
env HOST=10.0.0.3 \
PORT=8000 \
./run-app.sh
App
Exposes
10.0.0.3:8000
env HOST=10.0.0.4 \
PORT=8000 \
./run-app.sh
App
Exposes
10.0.0.4:8000
Load Balancer
$PUBLIC_IP



docker

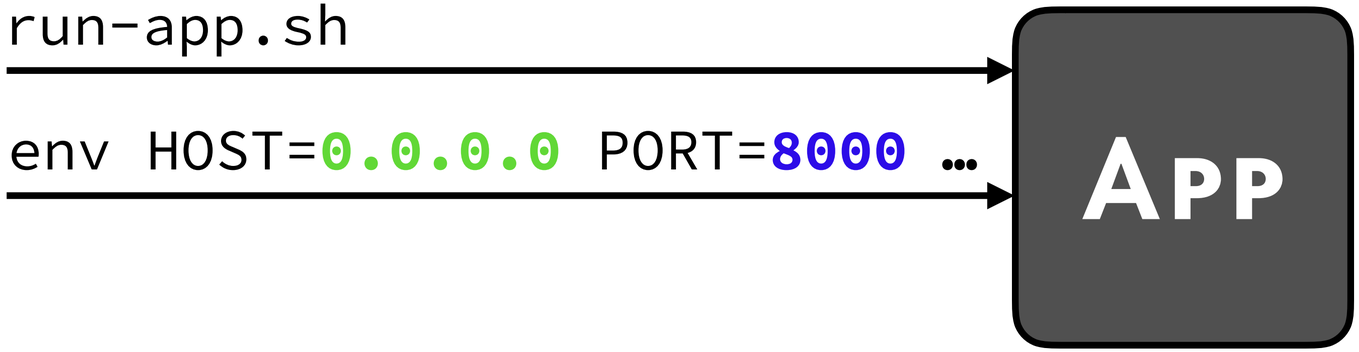


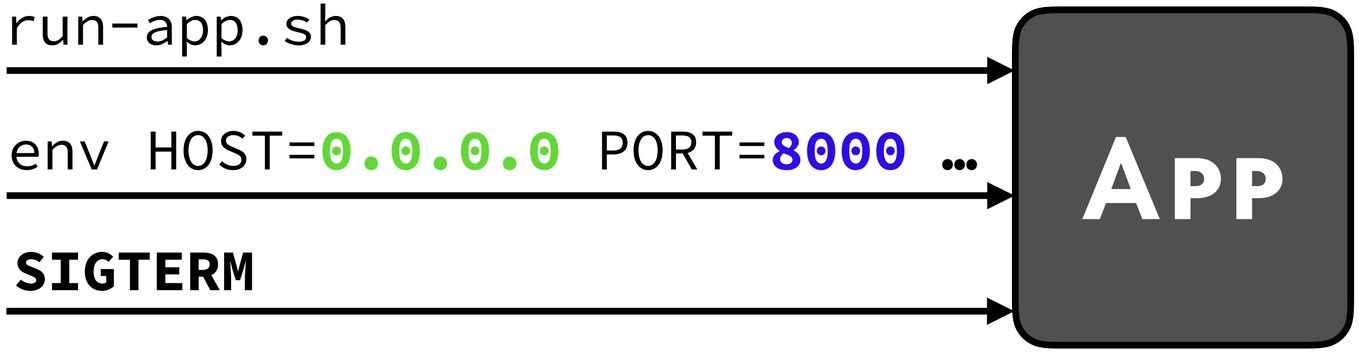
docker

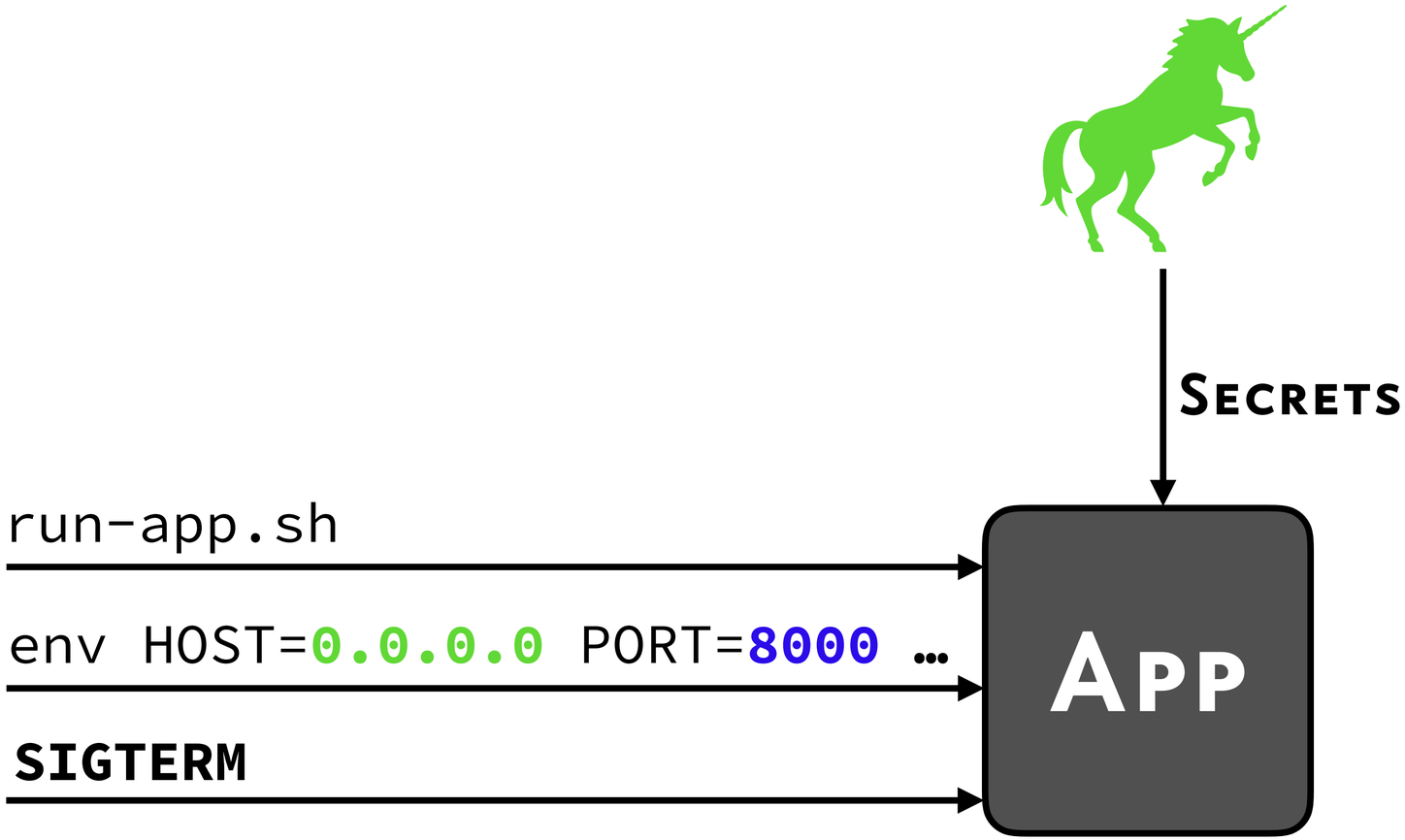


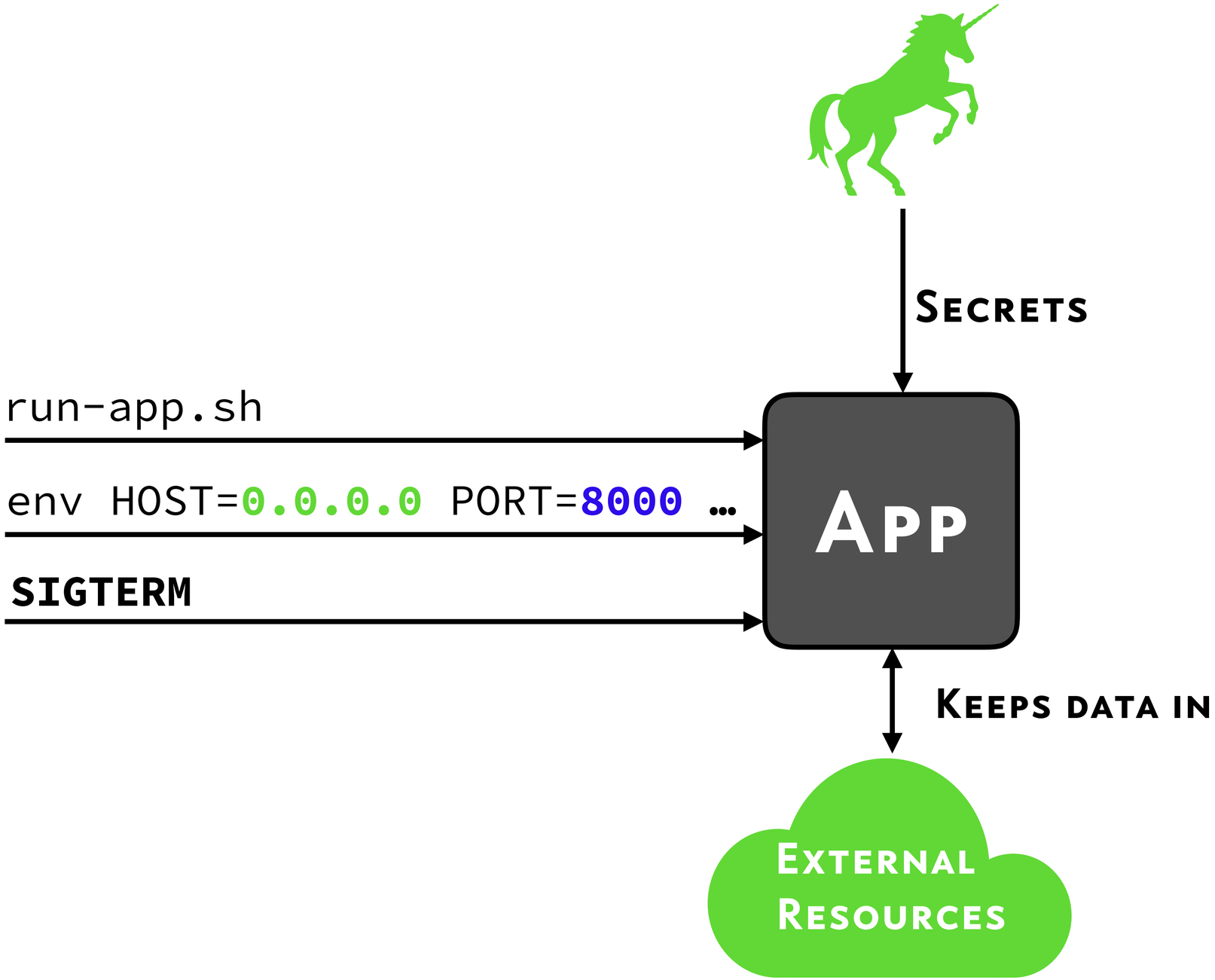

App

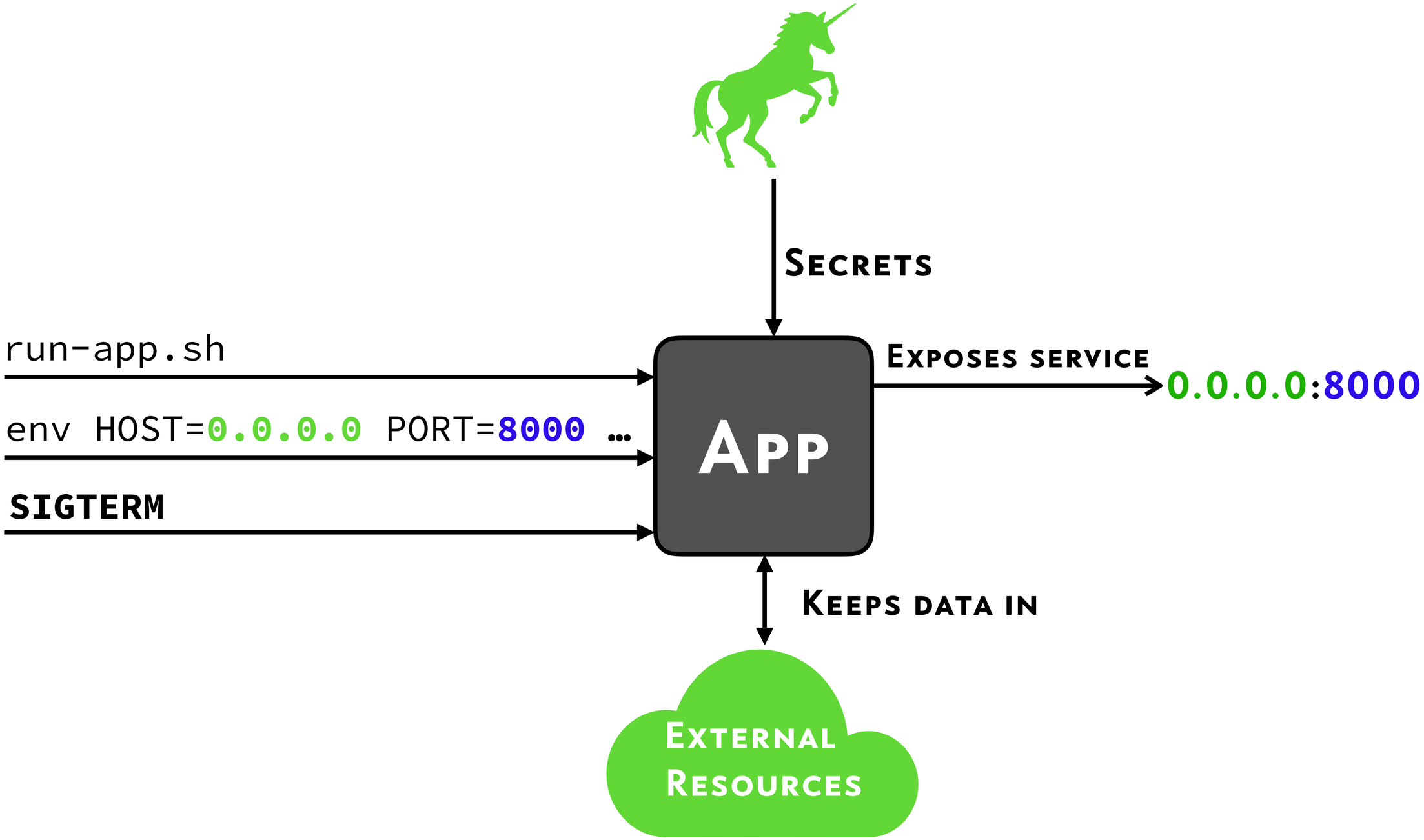

run-app.sh
App

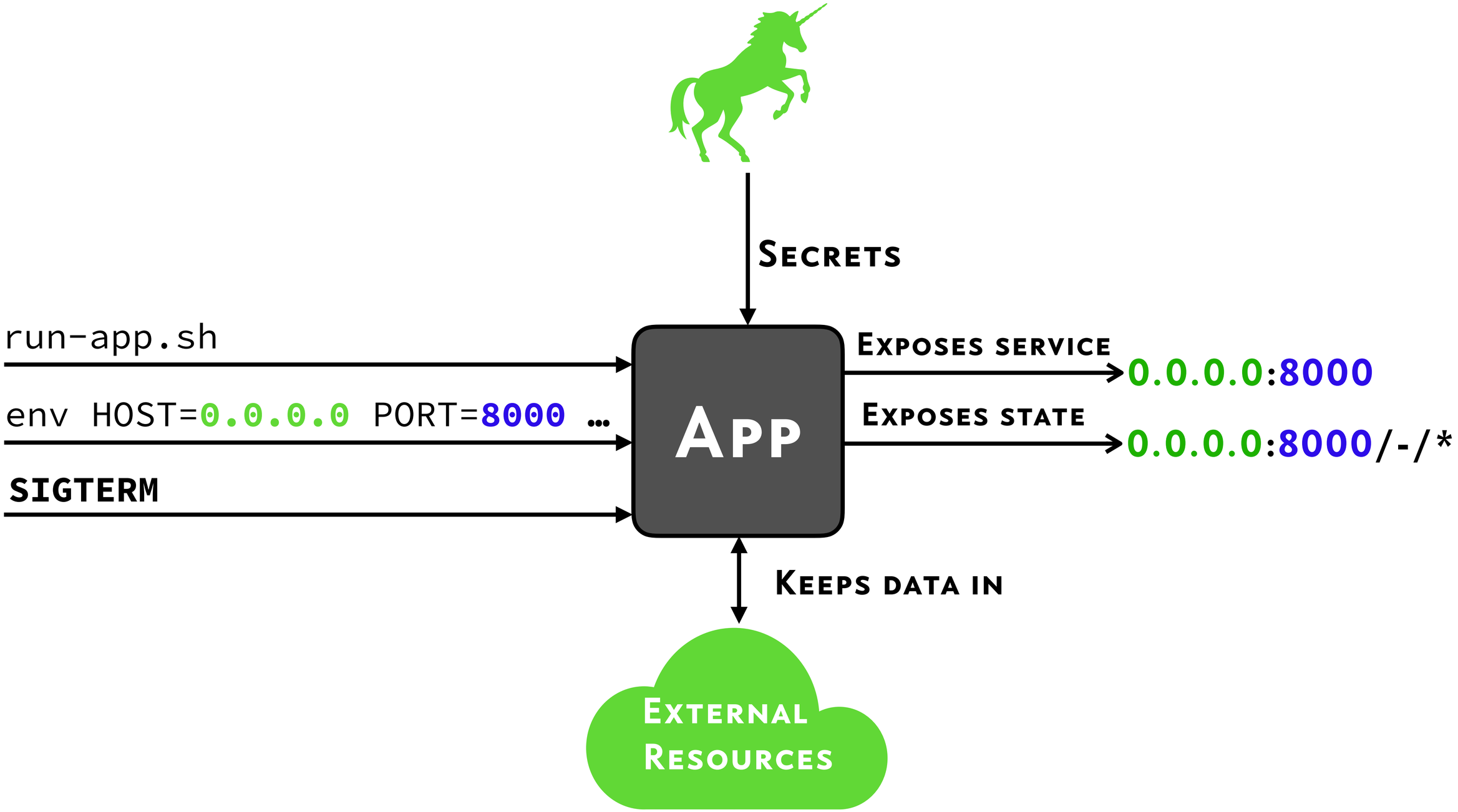

run-app.sh
env HOST=0.0.0.0 PORT=8000 …
App

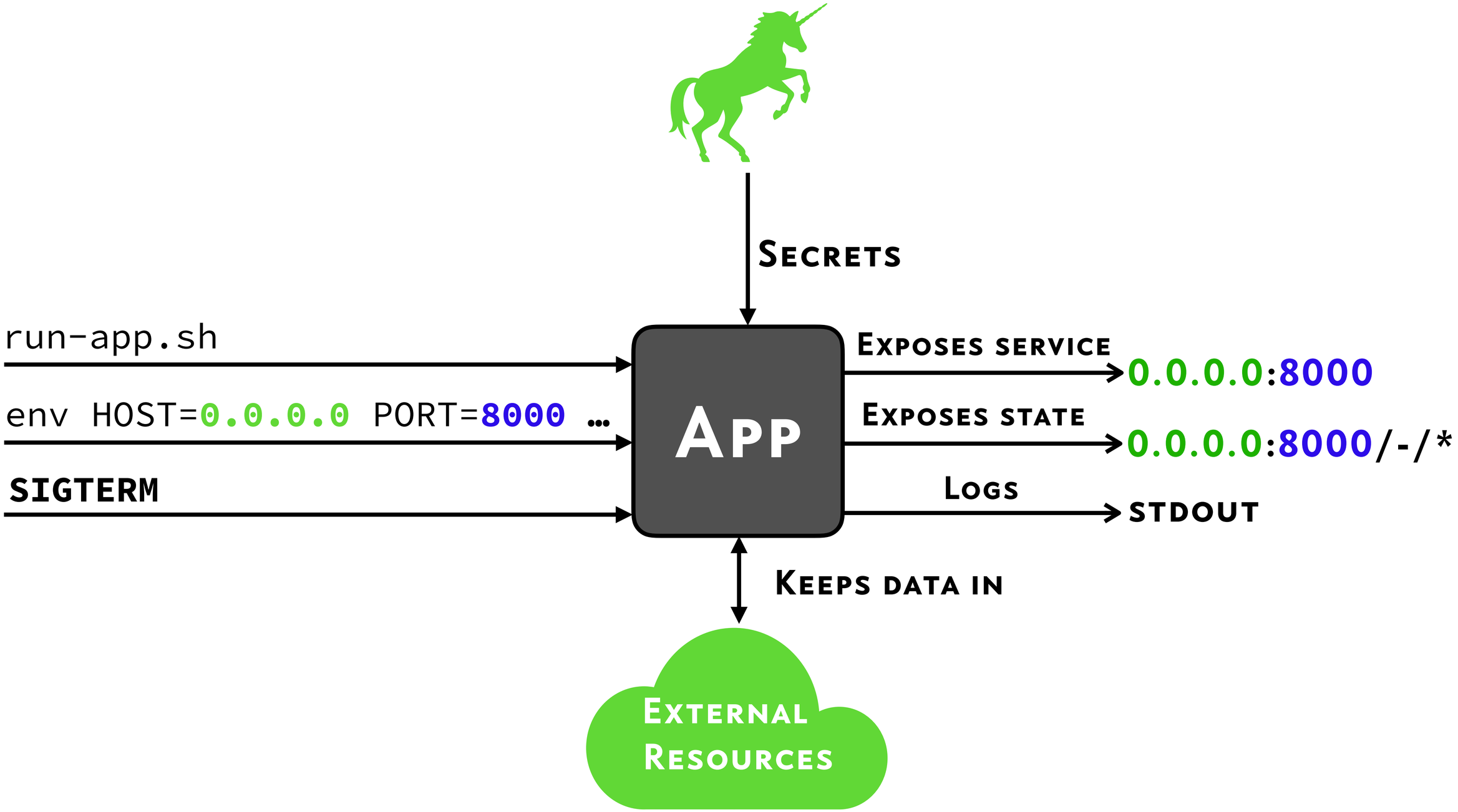

run-app.sh
env HOST=0.0.0.0 PORT=8000 …
SIGTERM
App


Secrets
run-app.sh
env HOST=0.0.0.0 PORT=8000 …
SIGTERM
App


Secrets
run-app.sh
env HOST=0.0.0.0 PORT=8000 …
SIGTERM
App
Keeps data in
External Resources


Secrets
run-app.sh
env HOST=0.0.0.0 PORT=8000 …
SIGTERM
App
Exposes service
0.0.0.0:8000
Keeps data in
External Resources


Secrets
run-app.sh
env HOST=0.0.0.0 PORT=8000 …
SIGTERM
App
Exposes service
0.0.0.0:8000
Exposes state
0.0.0.0:8000/-/*
Keeps data in
External Resources


Secrets
run-app.sh
env HOST=0.0.0.0 PORT=8000 …
SIGTERM
App
Exposes service
0.0.0.0:8000
Exposes state
0.0.0.0:8000/-/*
Logs
stdout
Keeps data in
External Resources

- Loose Coupling

- Loose Coupling
- **Separate I/O & Logic**

- Loose Coupling
- Separate I/O & Logic
- **Avoid Global State**

App Boundary

=

Just Another Boundary

Epilogue

Help us improve Python packaging – donate today!

Help Donate Log in Register

Find, install and publish Python packages with the Python Package Index

Search projects

Or browse projects

135,443 projects 941,541 releases 1,253,261 files 273,460 users

python™ Package Index

The Python Package Index (PyPI) is a repository of software for the Python programming language.

PyPI helps you find and install software developed and shared by the Python community. Learn about installing packages.

Package authors use PyPI to distribute their software. Learn how to package your Python code for PyPI.

OX.CX/DF

@HYNEK

VRMD.DE