

HOW TO WRITE DEPLOYMENT-FRIENDLY APPLICATIONS

HYNEK SCHLAWACK






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

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

```
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()
```

```
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


```
#!/bin/bash
```

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


#!/bin/bash

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


#!/bin/bash

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



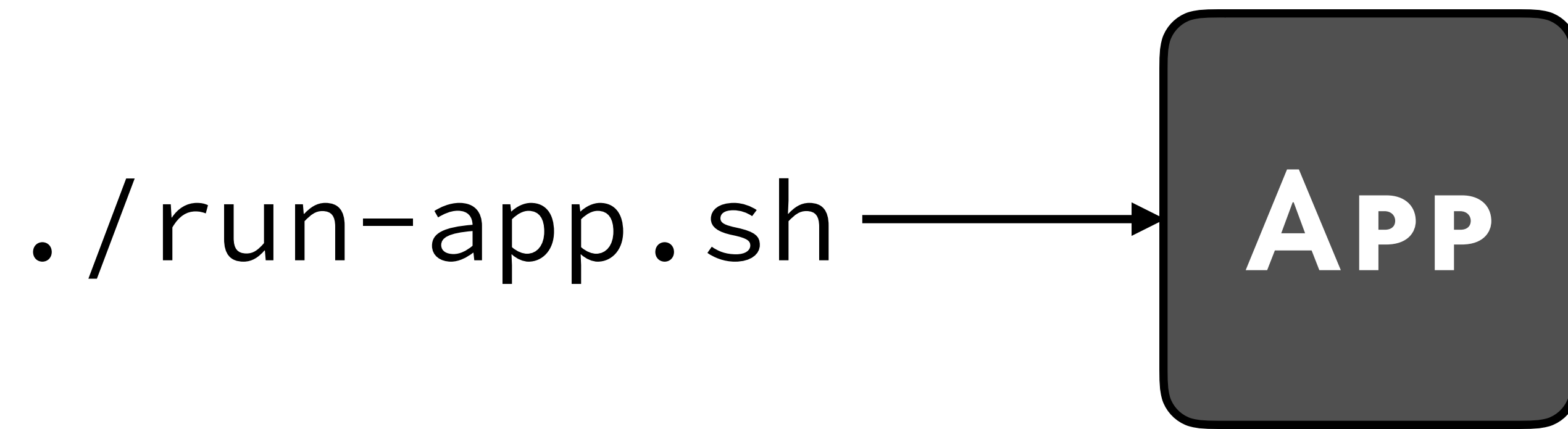
```
ExecStart=/app/run-app.sh
```

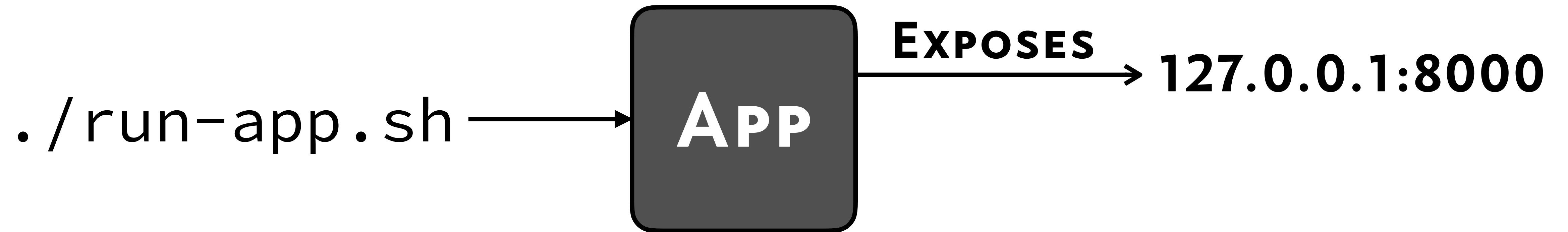


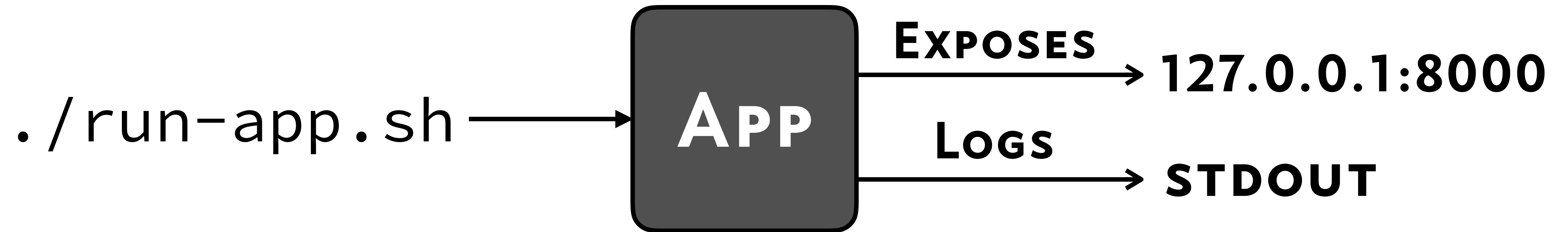
```
ENTRYPOINT ["/app/run-app.sh"]
```

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







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
```



```
ENV LOG_FORMAT=human
```

```
ENV LOG_LEVEL=INFO
```

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


DIRENV

ENVCONSUL

DIRENV

ETCDENV

ENVCONSUL

DIRENV

ETCDENV

os.environ

ENVCONSUL

DIRENV

ETCDENV

os.environ

ENVSUBST

CONFD

DIRENV

ENVCONSUL

ETCDENV

CONSUL-TEMPLATE

os.environ

ENVSUBST

```
$ 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,  
)
```

```
import environ
```

CONFIG.PY

```
@environ.config(prefix="APP")
```

```
class AppConfig:
```

```
    @environ.config
```

```
    class Log:
```

```
        level = environ.var()
```

```
        format = environ.var()
```

```
log = environ.group(Log)
```

```
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**


```
import environ
```

WSGI.PY

```
from .config import AppConfig
```

```
from .app_maker import make_app
```

```
app_cfg = environ.to_config(AppConfig)
```

```
application = make_app(app_cfg)
```

```
import environ
```

WSGI.PY

```
from .config import AppConfig
```

```
from .app_maker import make_app
```

```
app_cfg = environ.to_config(AppConfig)
```

```
application = make_app(app_cfg)
```

#!/bin/bash

RUN-APP.SH

```
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


```
class VaultSecrets:
```

SECRETS.PY

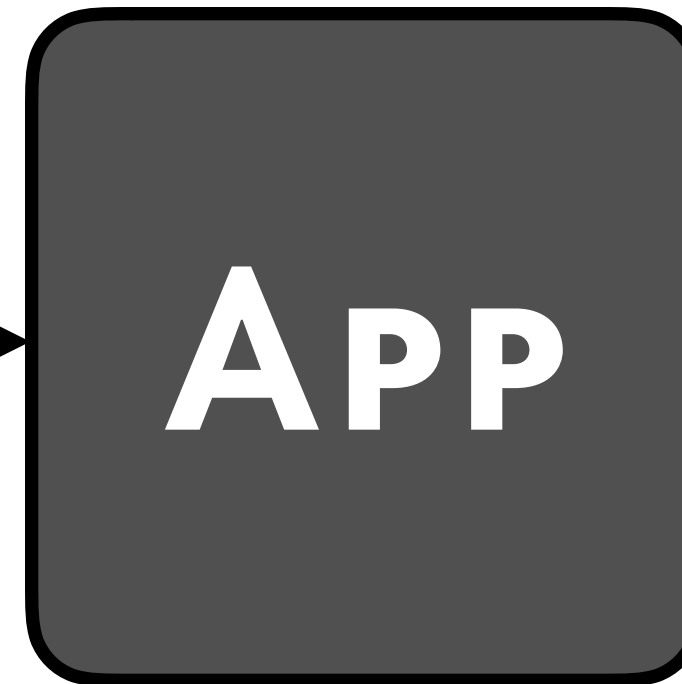
```
    # ...
```

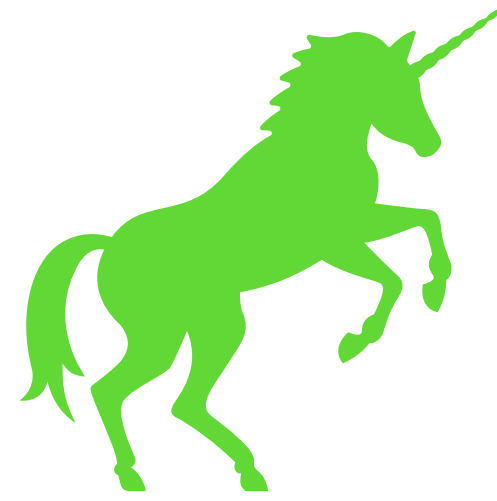
```
    def get_db_url(self):  
        return self.vault.read(  
            f"secret/{self.env}/app-name"  
        )["data"]["db_url"])
```

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




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

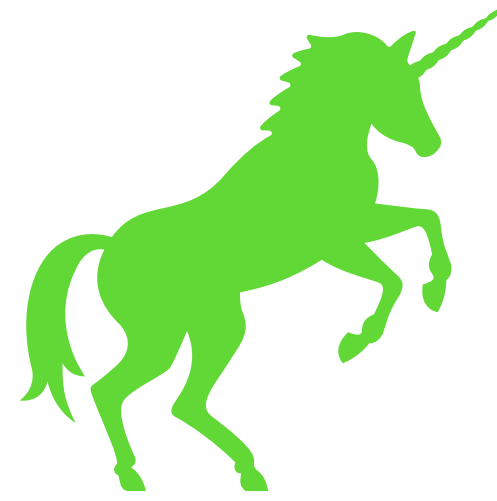




SECRETS

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

APP



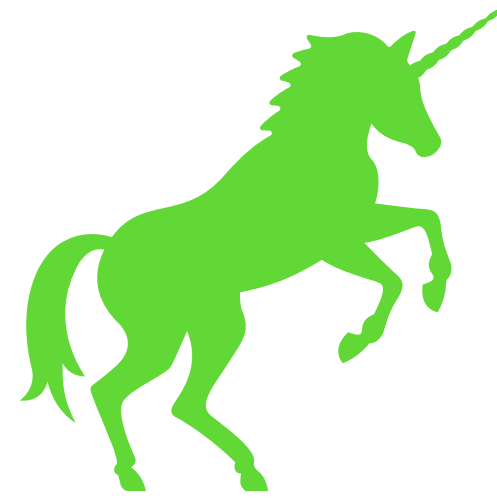
SECRETS

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

APP

EXPOSES

0.0.0.0:8000



SECRETS

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

APP

EXPOSES

0.0.0.0:8000

LOGS

@ INFO

STDOUT

env HOST=**127.0.0.1** \
PORT=**8000** \
./run-app.sh



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



EXPOSES

127.0.0.1:8000

env HOST=**127.0.0.1** \
PORT=**8001** \
./run-app.sh

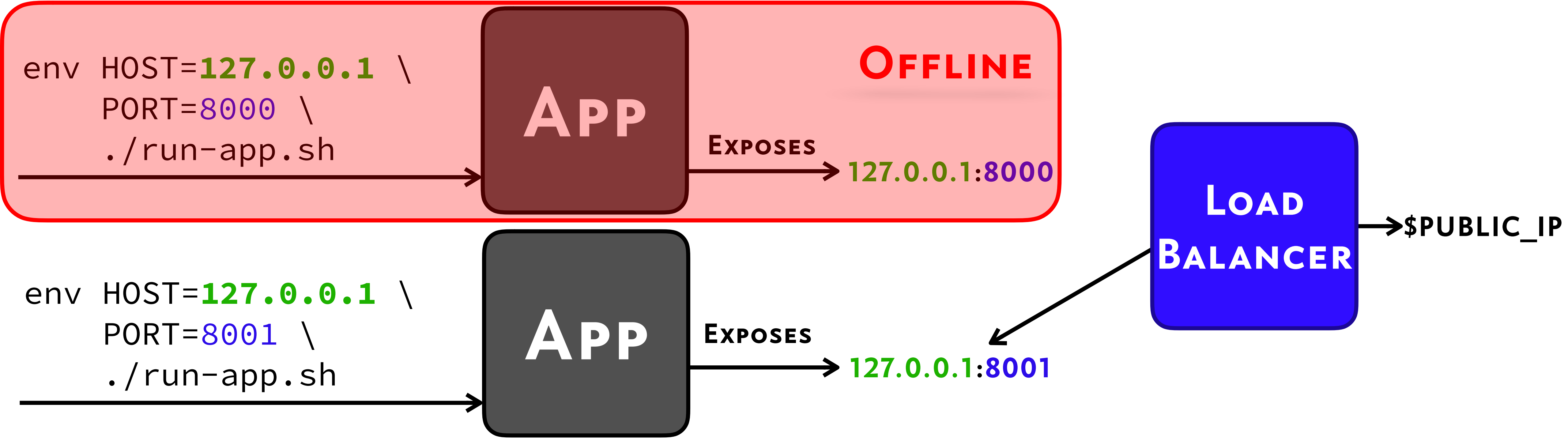


EXPOSES

127.0.0.1:8001



\$PUBLIC_IP



SIGTERM

env HOST=**127.0.0.1** \
PORT=**8000** \
./run-app.sh



EXPOSES

127.0.0.1:8000

env HOST=**127.0.0.1** \
PORT=**8001** \
./run-app.sh



EXPOSES

127.0.0.1:8001



\$PUBLIC_IP

env HOST=**127.0.0.1** \
PORT=**8000** \
./run-app.sh



EXPOSES

127.0.0.1:8000

env HOST=**127.0.0.1** \
PORT=**8001** \
./run-app.sh

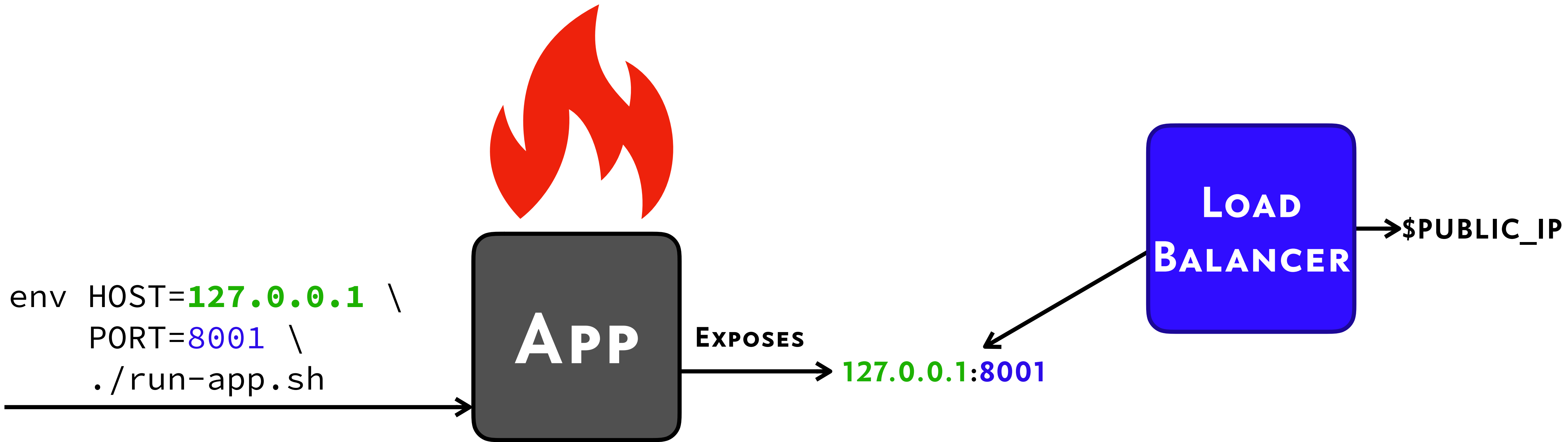


EXPOSES

127.0.0.1:8001



\$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")
```



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

```
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



EXPOSES

127.0.0.1:8000

env HOST=**127.0.0.1** \
PORT=**8001** \
./run-app.sh

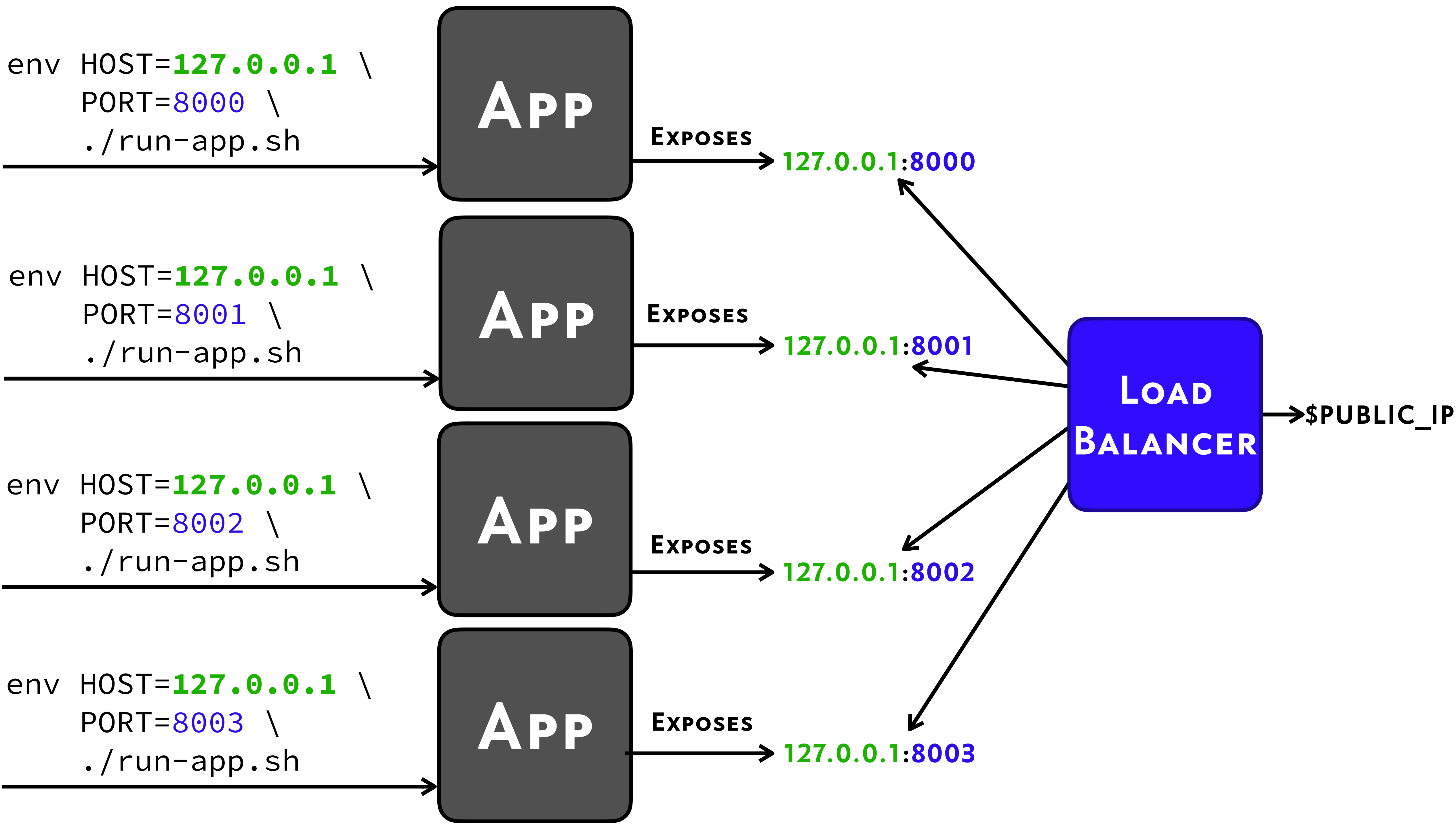


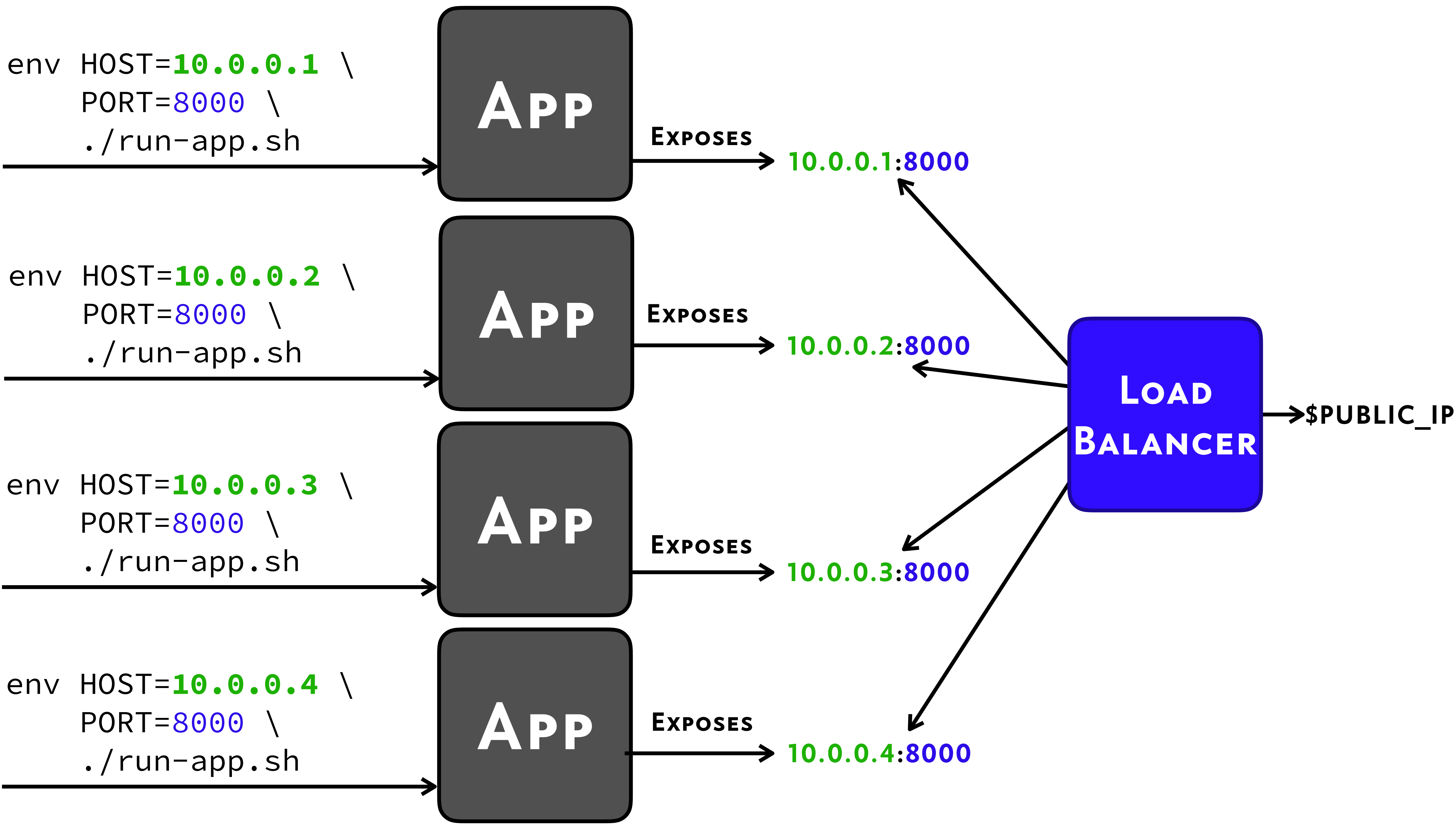
EXPOSES

127.0.0.1:8001

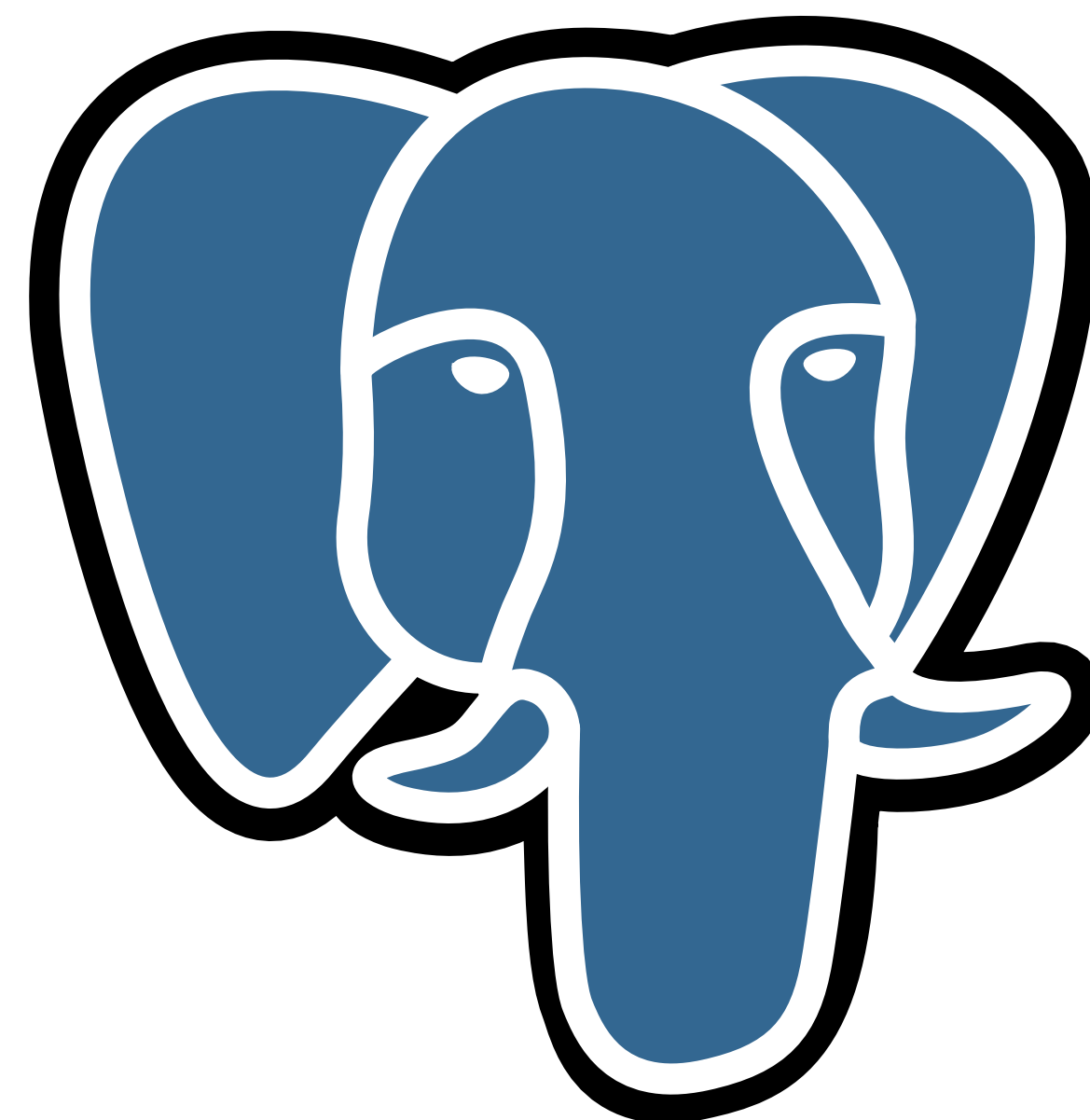


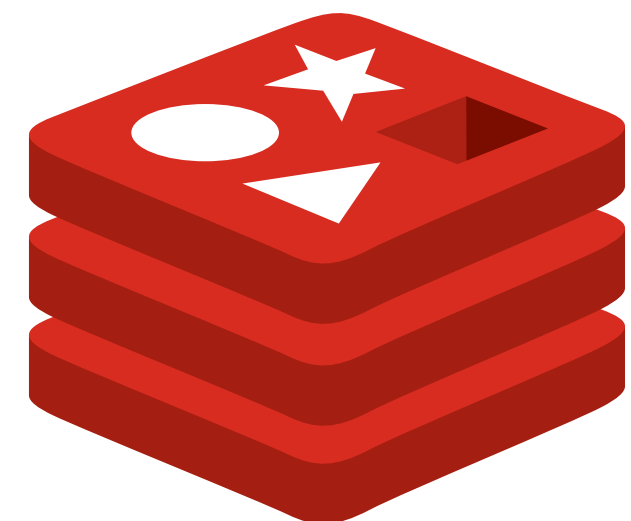
\$PUBLIC_IP



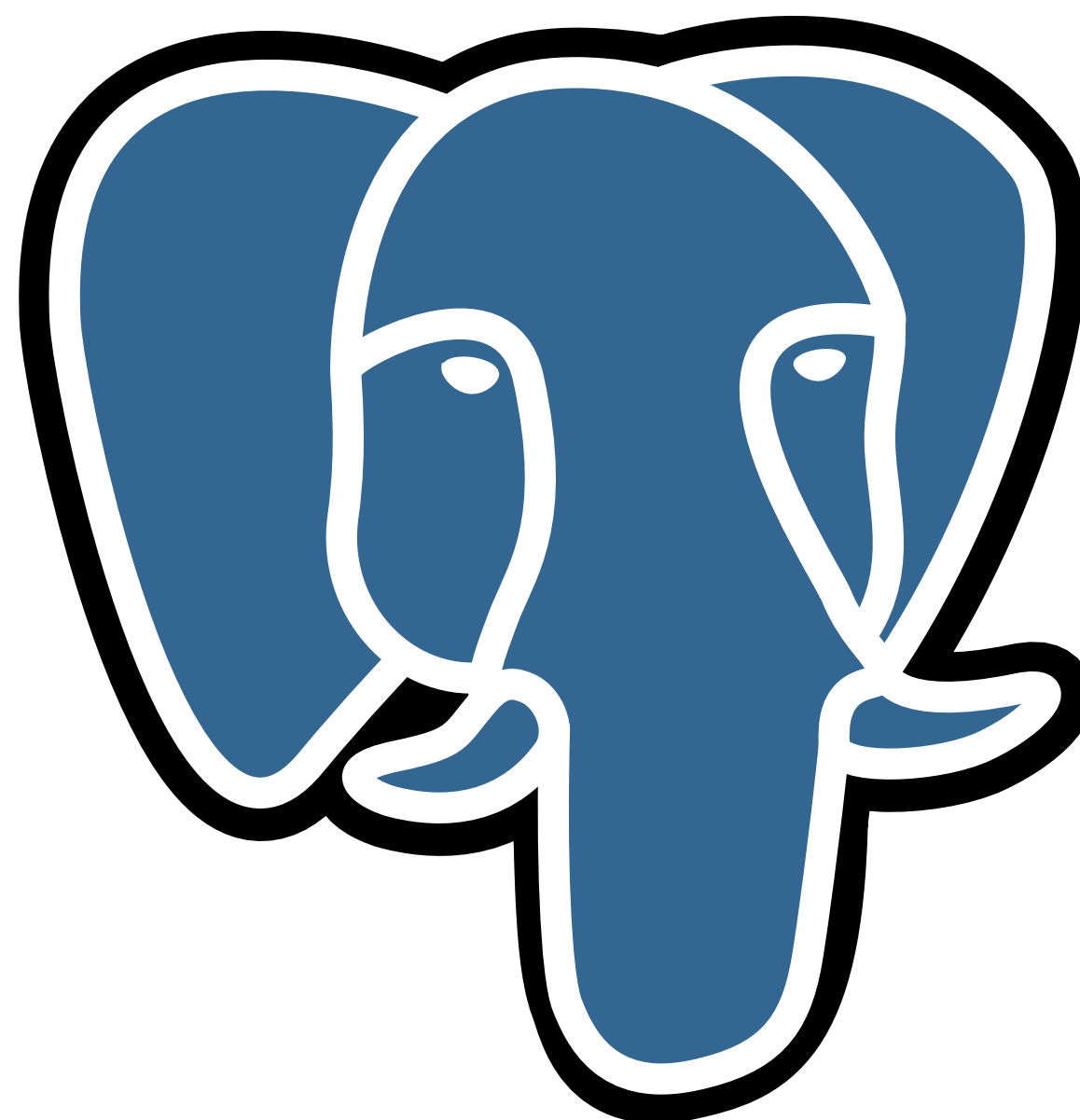


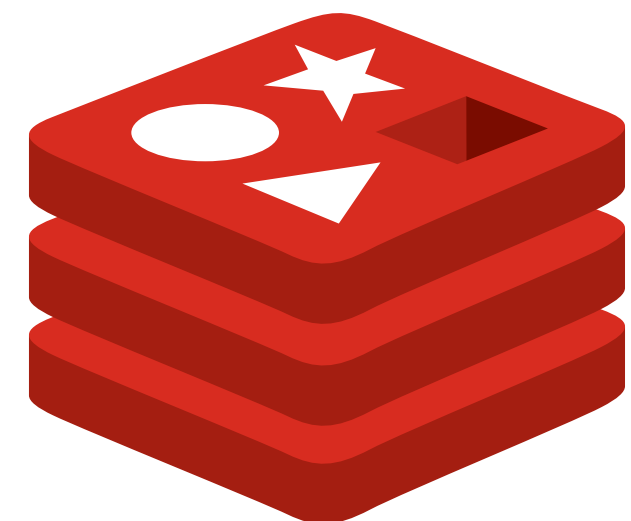




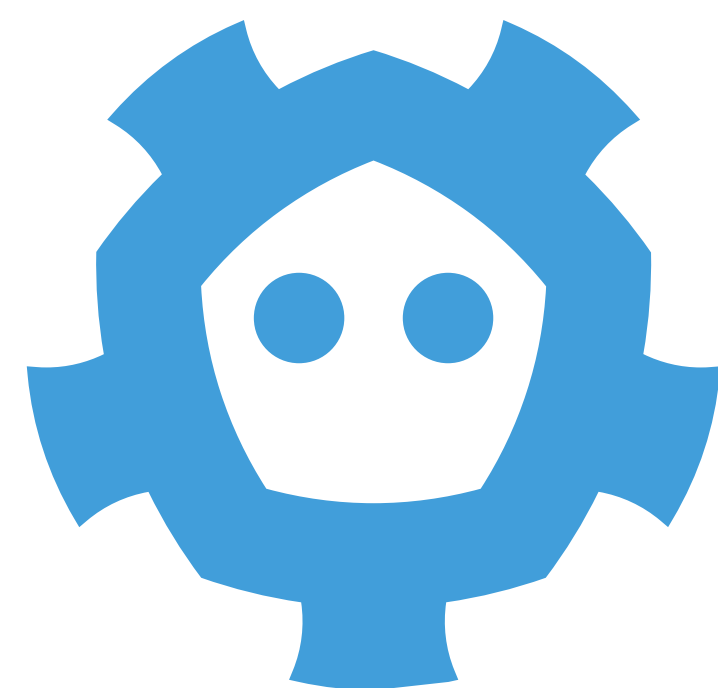
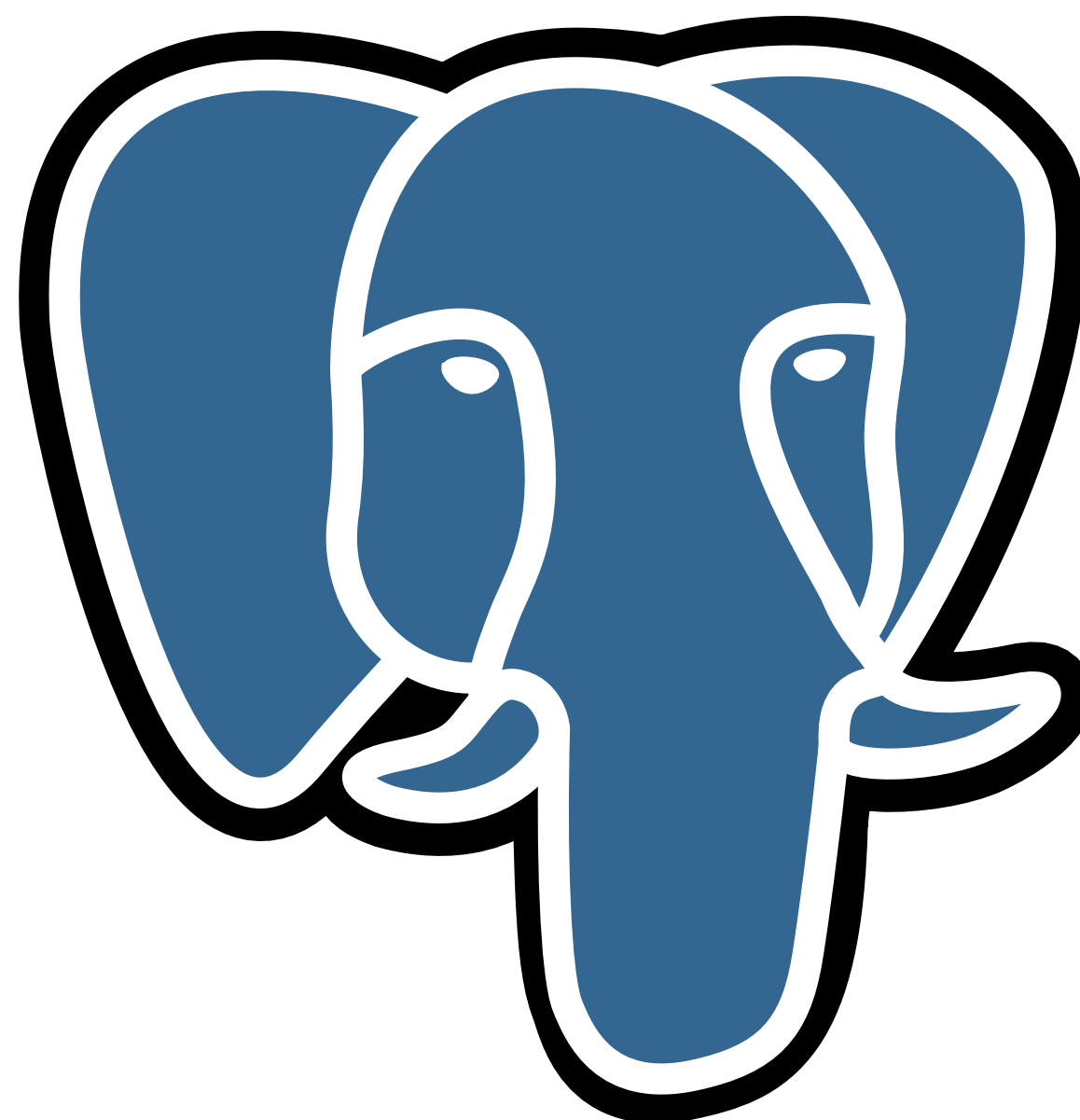


redis

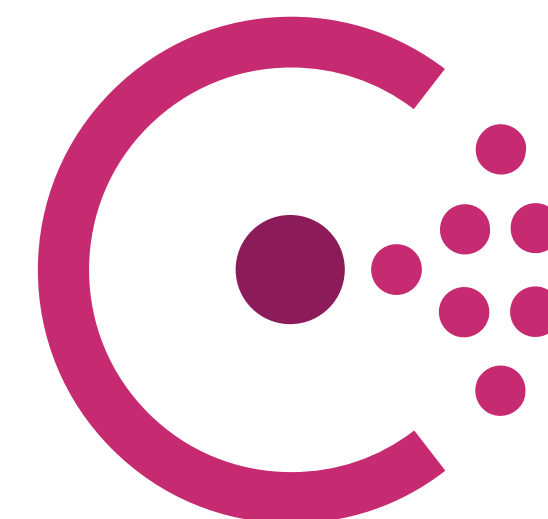




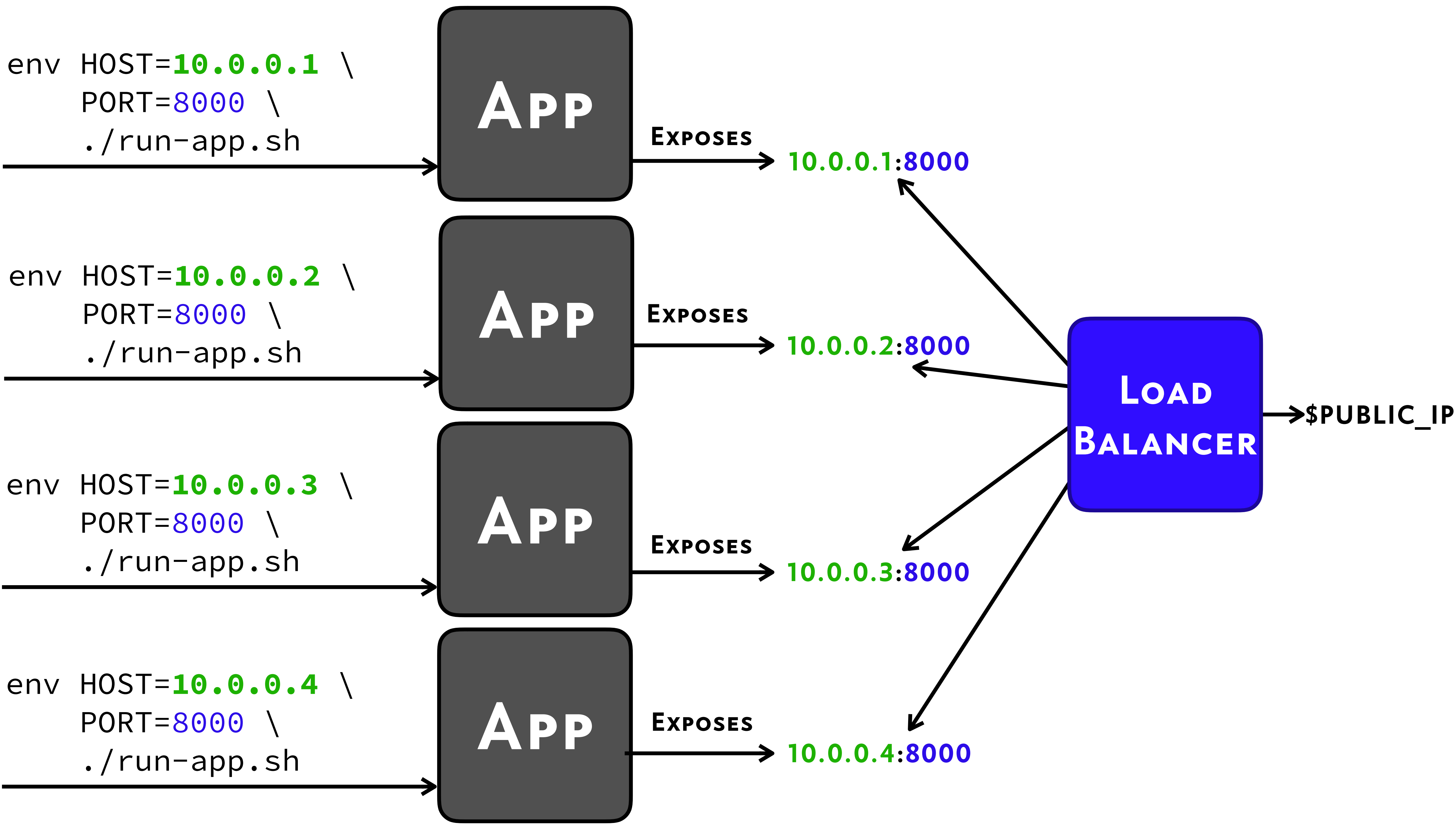
redis

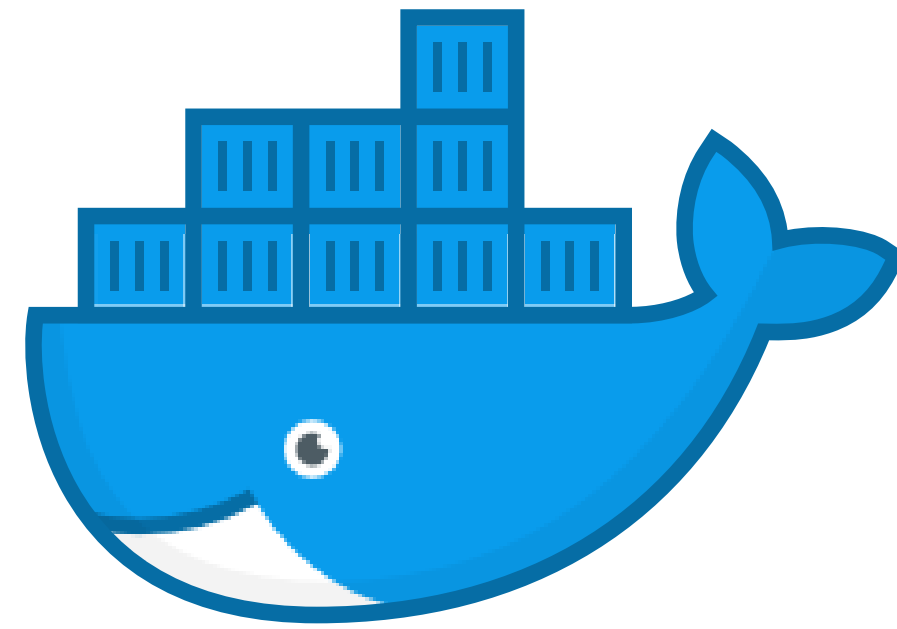


etcd

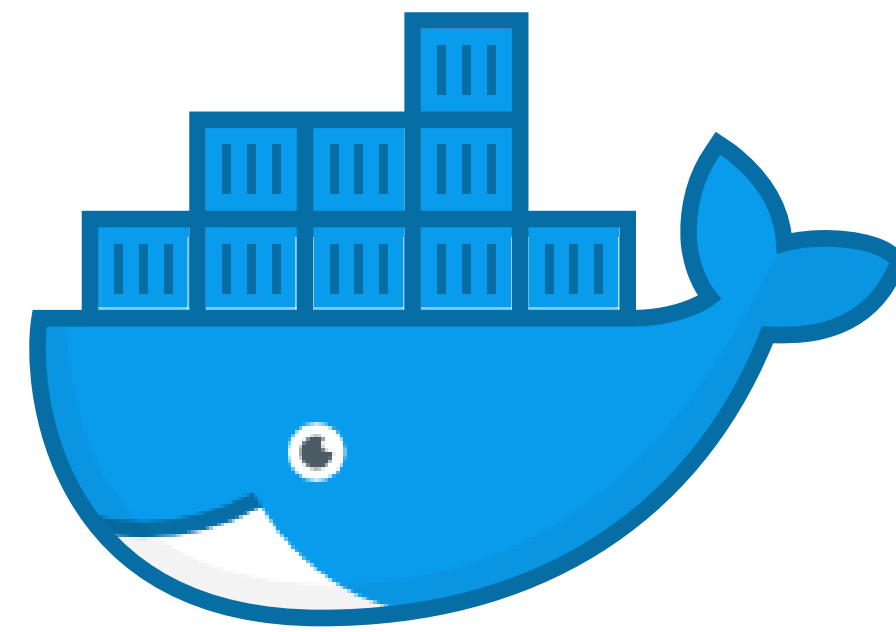
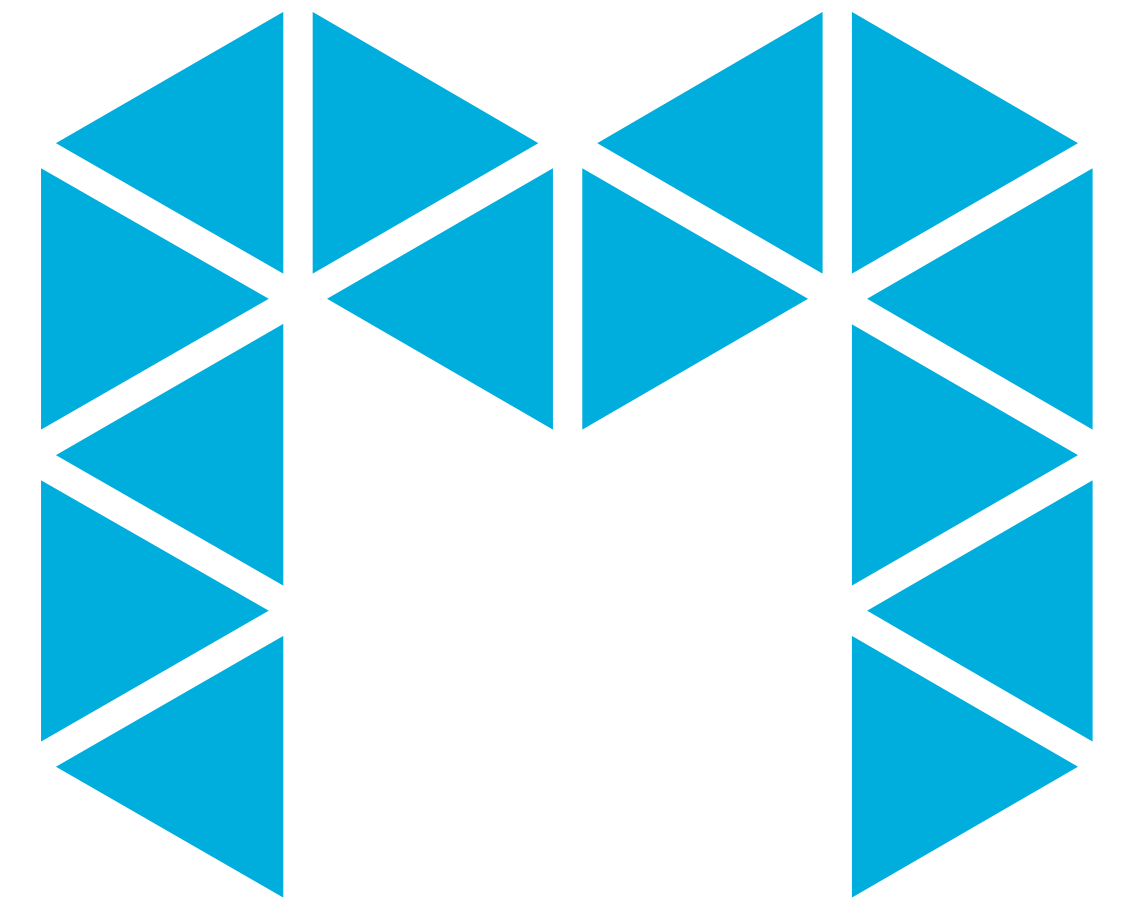


Consul

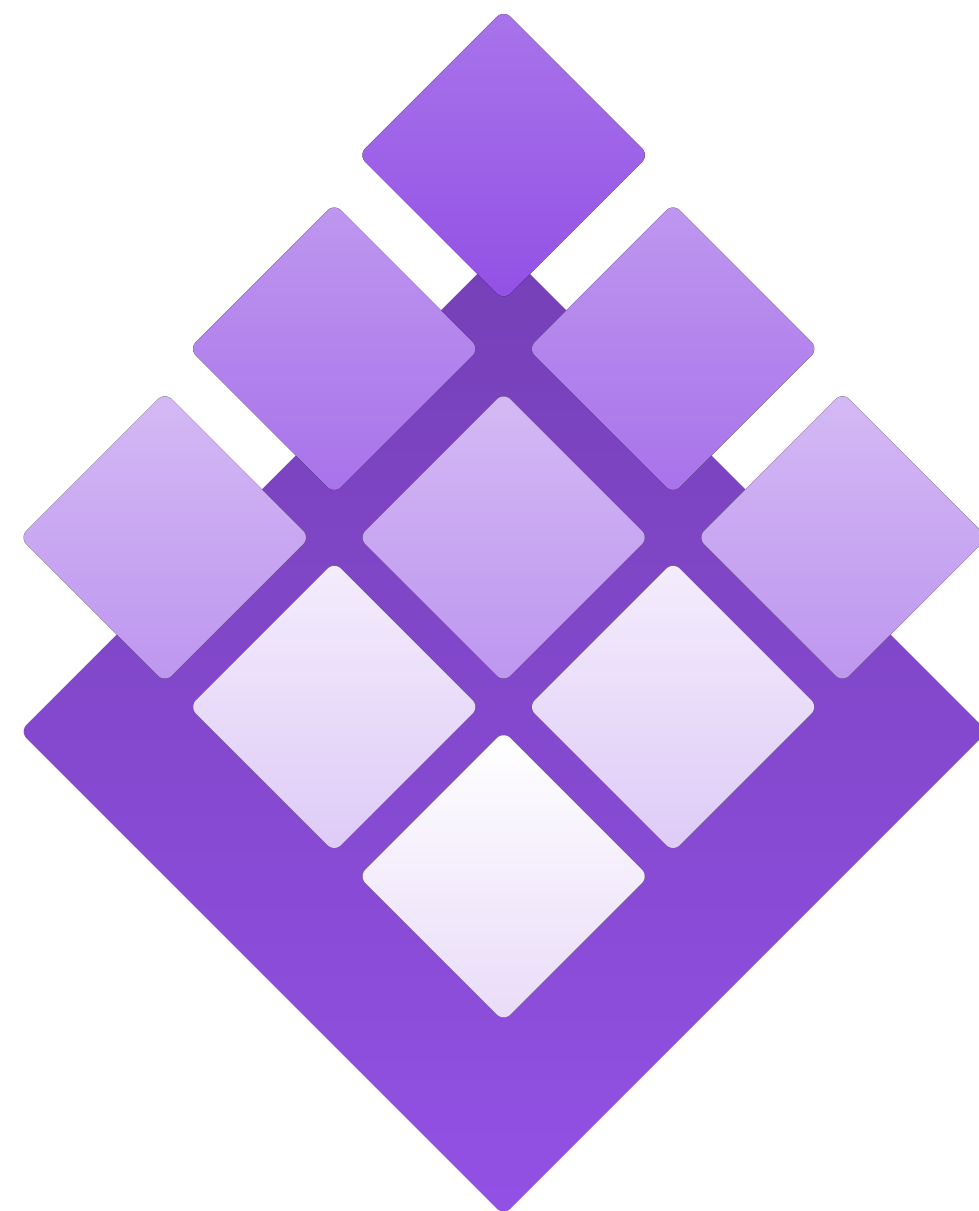




docker®

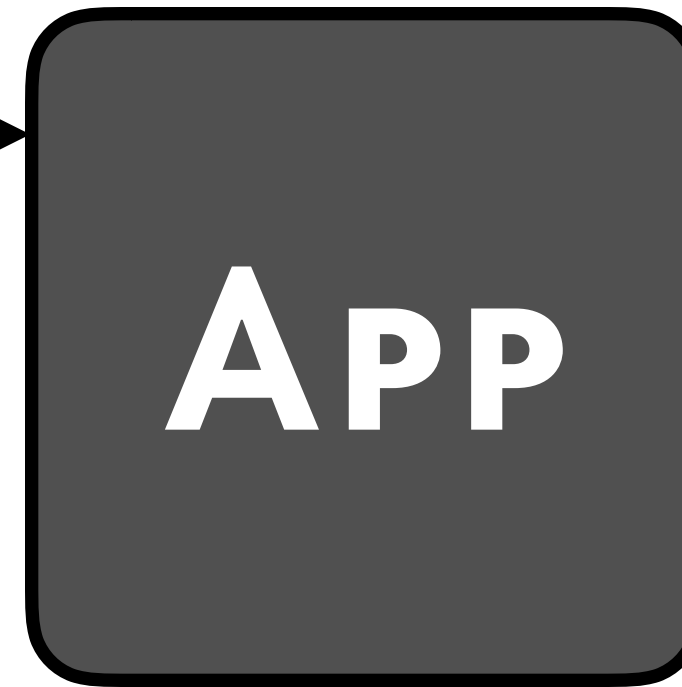


docker®





run-app.sh



APP

run-app.sh

env HOST=0.0.0.0 PORT=8000 ...

APP

run-app.sh

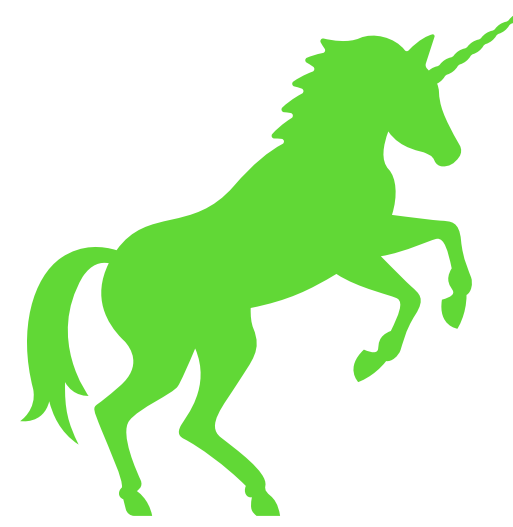
env HOST=0.0.0.0 PORT=8000 ...

SIGTERM

APP

```
graph LR; A[run-app.sh] --> D[APP]; B[env HOST=0.0.0.0 PORT=8000 ...] --> D; C[SIGTERM] --> D;
```

The diagram illustrates three inputs being sent to an application component. On the left, three lines of text are stacked vertically. Each line is underlined and has a horizontal arrow pointing from its right end to the left side of a dark gray rounded rectangle. The rectangle is labeled 'APP' in white capital letters. The first line is 'run-app.sh'. The second line is 'env HOST=0.0.0.0 PORT=8000 ...', where the IP address is green and the port is blue. The third line is 'SIGTERM' in bold black text.



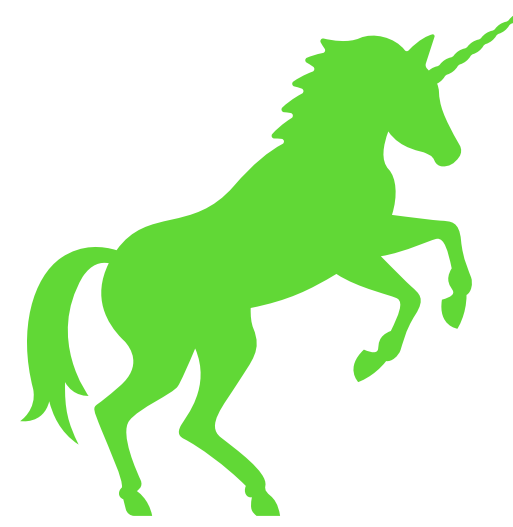
SECRETS

`run-app.sh`

`env HOST=0.0.0.0 PORT=8000 ...`

SIGTERM

APP



SECRETS

`run-app.sh`

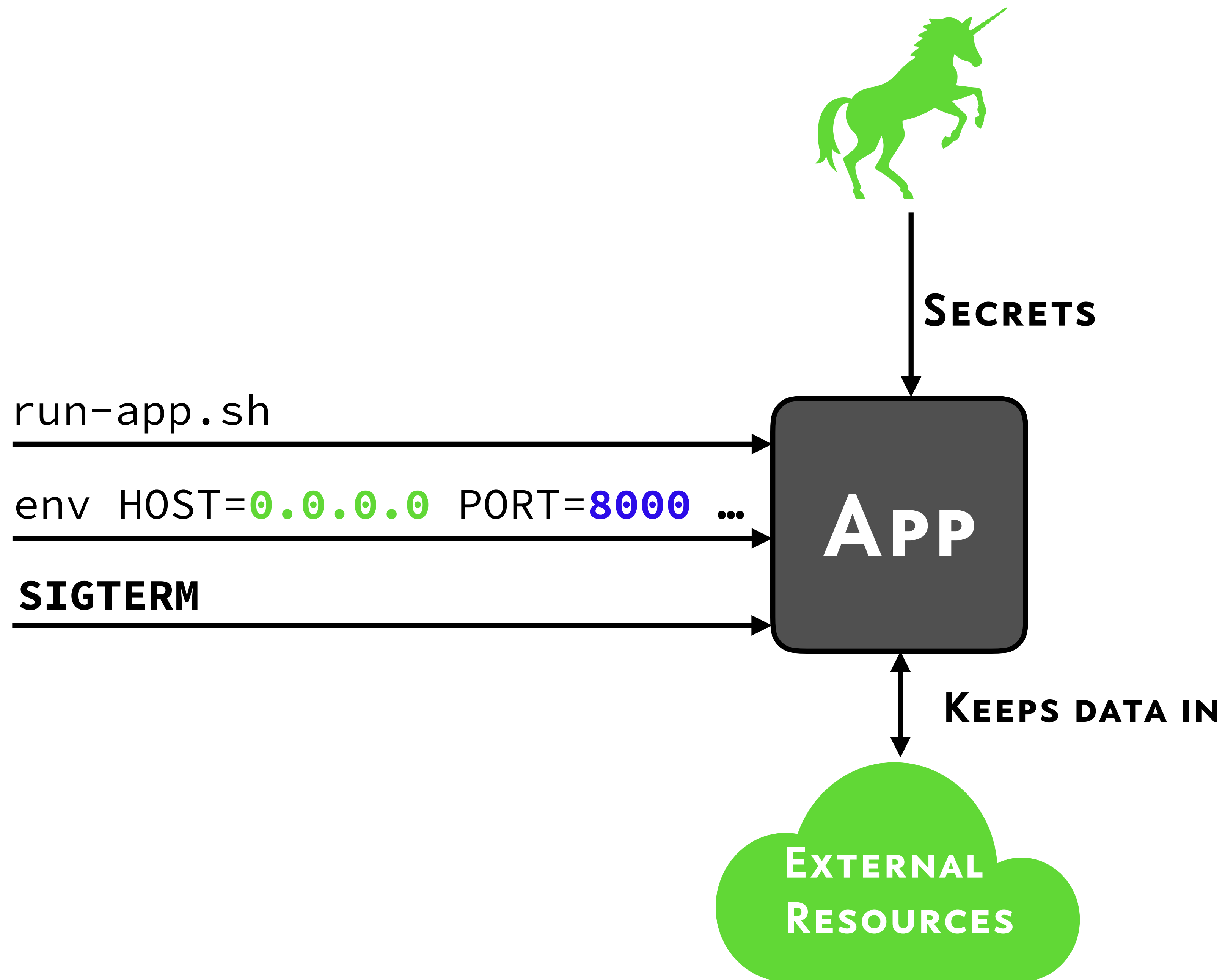
`env HOST=0.0.0.0 PORT=8000 ...`

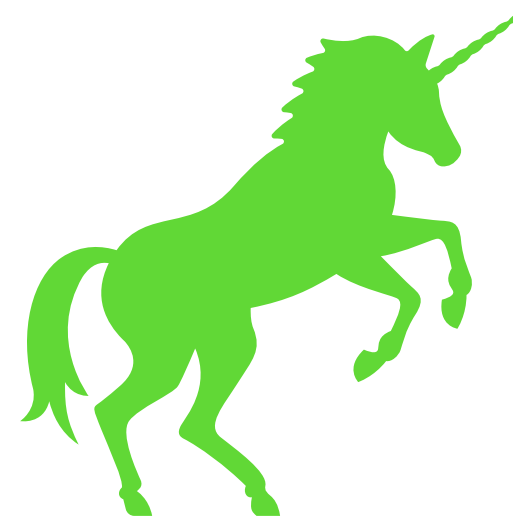
SIGTERM

APP

KEEPS DATA IN

**EXTERNAL
RESOURCES**





SECRETS

`run-app.sh`

`env HOST=0.0.0.0 PORT=8000 ...`

SIGTERM

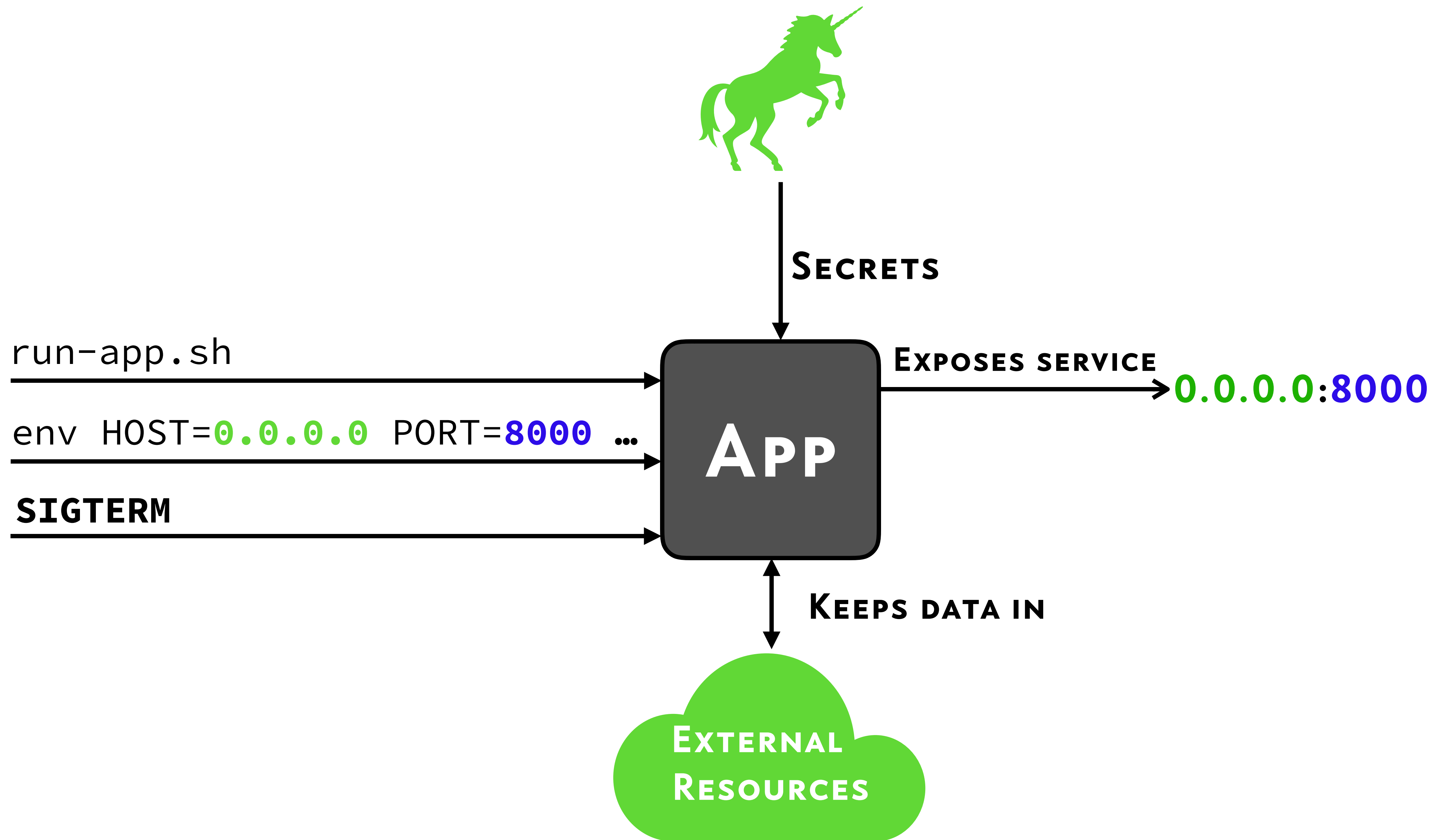
APP

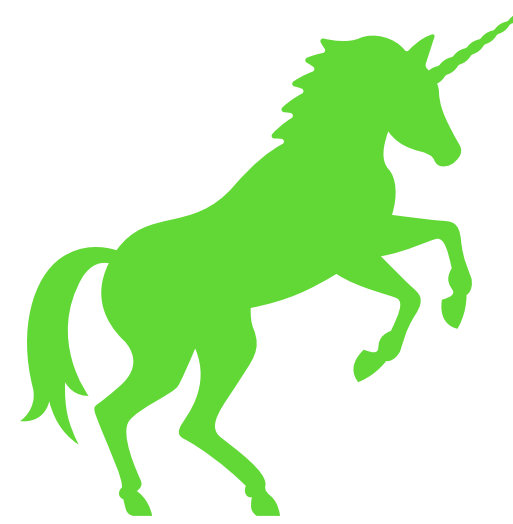
EXPOSES SERVICE

`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

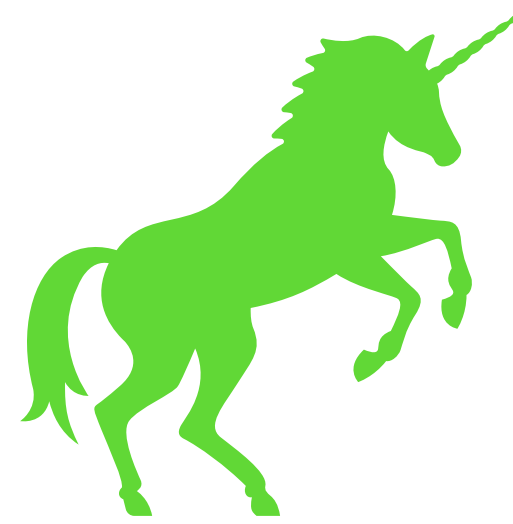
`0.0.0.0:8000`

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



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



Or [browse projects](#)

135,443 projects

941,541 releases

1,253,261 files

273,460 users



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

