Now that PostgreSQL is properly configured, we need to create a configuration file to inform Vault that its storage backend will be the Vault database inside the `postgresql` container. Let's do that by defining the following configuration file named config.hcl.

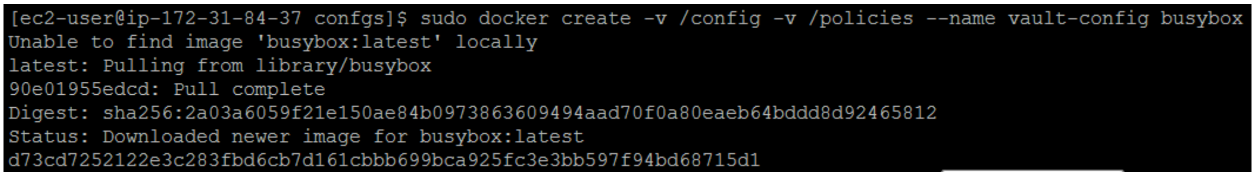
```
{
  "backend": {"postgresql": {"connection_url": "postgres://postgres: postgres@postgres-7ff9df5765-
nqzbk:5432/postgresdb?sslmode=disable"}},
  "listener": {"tcp": {"address": "0.0.0.0:8200", "tls_disable": 1}}
}
```

Using Vault we can make use of the Access Control Policies – ACLs – to define different policies to allow or deny access to specific secrets. Before proceeding, let's define a simple file that will be used to allow read-only access to each secret contained inside *secret/web* path to any authenticated user or service that will be identified as part of that policy:

```
# web-policy.hcl
path "secret/web/*" {
  policy = "read"
}
```

Both files will be stored inside a Docker data container to be easily accessible from other linked containers. Let's create the container by executing:

```
$ docker create -v /config -v /policies --name vault-config
busybox
```



```
[ec2-user@ip-172-31-84-37 confgs]$ sudo docker create -v /config -v /policies --name vault-config busybox
Unable to find image 'busybox:latest' locally
latest: Pulling from library/busybox
90e01955edcd: Pull complete
Digest: sha256:2a03a6059f21e150ae84b0973863609494aad70f0a80eaeb64bddd8d92465812
Status: Downloaded newer image for busybox:latest
d73cd7252122e3c283fbd6cb7d161cbbb699bca925fc3e3bb597f94bd68715d1
```

Next, we will copy both of the files inside it:

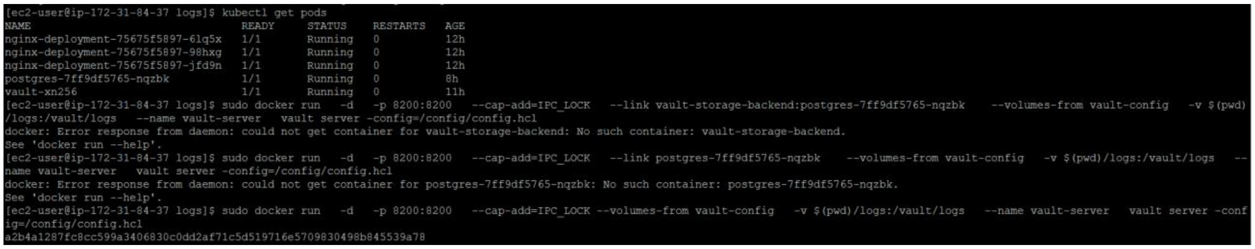
```
$ docker cp config.hcl vault-config:/config/
$ docker cp web-policy.hcl vault-config:/policies/
```

Since we want to make use of Vault’s auditing capabilities and we want to make logs persistent, we will store them in a local folder on the host and then mount it in Vault’s container. Let’s create the local folder:

```
mkdir logs
```

Finally, we can start our Vault server by launching a container named *vault-server:*

```
docker run \
  -d \
  -p 8200:8200 \
  --cap-add=IPC_LOCK \
  --link vault-storage-backend:storage-backend  \
  --volumes-from vault-config \
  -v $(pwd)/logs:/vault/logs \
  --name vault-server \
  vault server -config=/config/config.hcl
```



```
[ec2-user@ip-172-31-84-37 logs]$ kubectl get pods
NAME                                READY     STATUS    RESTARTS   AGE
nginx-deployment-75675f5897-6lq5x   1/1       Running   0          12h
nginx-deployment-75675f5897-98hxg   1/1       Running   0          12h
nginx-deployment-75675f5897-jfd9n   1/1       Running   0          12h
postgres-7ff9df5765-nqzbk           1/1       Running   0          8h
vault-xn256                         1/1       Running   0          11h
[ec2-user@ip-172-31-84-37 logs]$ sudo docker run   -d   -p 8200:8200   --cap-add=IPC_LOCK   --link vault-storage-backend:postgres-7ff9df5765-nqzbk    --volumes-from vault-config   -v $(pwd)
/logs:/vault/logs   --name vault-server   vault server -config=/config/config.hcl
docker: Error response from daemon: could not get container for vault-storage-backend: No such container: vault-storage-backend.
See 'docker run --help'.
[ec2-user@ip-172-31-84-37 logs]$ sudo docker run   -d   -p 8200:8200   --cap-add=IPC_LOCK   --link postgres-7ff9df5765-nqzbk    --volumes-from vault-config   -v $(pwd)/logs:/vault/logs   --
name vault-server   vault server -config=/config/config.hcl
docker: Error response from daemon: could not get container for postgres-7ff9df5765-nqzbk: No such container: postgres-7ff9df5765-nqzbk.
See 'docker run --help'.
[ec2-user@ip-172-31-84-37 logs]$ sudo docker run   -d   -p 8200:8200   --cap-add=IPC_LOCK --volumes-from vault-config   -v $(pwd)/logs:/vault/logs   --name vault-server   vault server -conf
ig=/config/config.hcl
a2b4a1287fc8cc599a3406830c0dd2af71c5d519716e5709830498b845539a78
```