3. Export pre-JSONified data to a .sql dump

D:\liveability\data>pg_dump -U postgres -h localhost -W -t li_map_json_hard li_vic > ligres-hard-db.sql
 - this is only 29mb in size, which is interesting…

Example dumps and syncs etc:

Dev dump
pg_dump -U postgres -h localhost -W -t li_map_json_h_mini li_vic > ligres-h-mini-db.sql
psql --host=ligres.postgres.database.azure.com --port=5432 --username=yodel@ligres --dbname=li_hard -f ligres-h-mini-db.sql

pg_dump -U postgres -h localhost -W -t li_map_json_sa1_min_soft li_vic > li_map_json_sa1_min_soft.sql
pg_dump -U postgres -h localhost -W -t li_map_json_ssc_min_soft li_vic > li_map_json_ssc_min_soft.sql

pg_dump -U postgres -h localhost -W -t clean_li_map_json_sa1_min_soft li_vic > clean_li_map_json_sa1_min_soft.sql
pg_dump -U postgres -h localhost -W -t clean_li_map_json_ssc_min_soft li_vic > clean_li_map_json_ssc_min_soft.sql

Connection examples:

```
psql --host=ligres.postgres.database.azure.com --port=5432 --username=yodel@ligres --dbname=postgres
```

```
 CREATE DATABASE li_soft
  WITH OWNER = yodel
  ENCODING = 'UTF8'
  CONNECTION LIMIT = -1
  TEMPLATE template0;
\c li_soft
CREATE EXTENSION POSTGIS;
```

psql --host=ligres.postgres.database.azure.com --port=5432 --username=yodel@ligres --dbname=li_soft -f li_map_json_sa1_min_soft.sql
psql --host=ligres.postgres.database.azure.com --port=5432 --username=yodel@ligres --dbname=li_soft -f li_map_json_ssc_min_soft.sql

psql --host=ligres.postgres.database.azure.com --port=5432 --username=yodel@ligres --dbname=li_soft -f clean_li_map_json_sa1_min_soft.sql
psql --host=ligres.postgres.database.azure.com --port=5432 --username=yodel@ligres --dbname=li_soft -f clean_li_map_json_ssc_min_soft.sql

psql --host=ligres.postgres.database.azure.com --port=5432 --username=yodel@ligres --dbname=postgres

GRANT CONNECT ON DATABASE li_soft TO spatial;
GRANT SELECT ON ALL TABLES IN SCHEMA public TO spatial;
REVOKE CONNECT ON DATABASE postgres FROM spatial;

GRANT SELECT ON clean_li_map_json_sa1_min_soft TO spatial;
GRANT SELECT ON clean_li_map_json_ssc_min_soft TO spatial;

4. Follow rest of process

Note the key is the usual reason hashed with an answer of upper quints

log in to psql	psql --host=ligres.postgres.database.azure.com --port=5432 --username=yodel@ligres --dbname=postgres
Create db	CREATE DATABASE li_hard WITH OWNER = yodel ENCODING = 'UTF8' CONNECTION LIMIT = -1 TEMPLATE template0;
Connect	\c li_hard
enable postgis extension	CREATE EXTENSION POSTGIS;
restore table	psql --host=ligres.postgres.database.azure.com --port=5432 --username=yodel@ligres --dbname=li_hard -f ligres-hard-db.sql The restoration of the db may report issues with roles not existing , but according to this post (https://stackoverflow.com/questions/17153008/import-postgres-database-without-roles) non-existant roles will be transferred to the importer.
Check the above worked	psql --host=ligres.postgres.database.azure.com --port=5432 --username=yodel@ligres --dbname=li_hard \dt
Create read only user	CREATE USER spatial WITH LOGIN NOSUPERUSER NOCREATEDB NOCREATEROLE INHERIT

	NOREPLICATION CONNECTION LIMIT -1 PASSWORD 'sp$nUggy'; GRANT CONNECT ON DATABASE li_hard TO spatial; GRANT SELECT ON ALL TABLES IN SCHEMA public TO spatial; REVOKE CONNECT ON DATABASE postgres FROM spatial;
add firewall exceptions	az postgres server firewall-rule create --resource-group liveability --server-name ligres --start-ip-address 180.181.209.62 --end-ip-address 180.181.209.62 --name testConnect2
	az postgres server firewall-rule create --resource-group liveability --server-name ligres --start-ip-address 52.255.61.4 --end-ip-address 52.255.61.4 --name testConnect1
	az postgres server firewall-rule create --resource-group liveability --server-name ligres --start-ip-address 192.168.1.2 --end-ip-address 192.168.1.2 --name carlHome
	etc (the first two are for accessing via azure --- this may need some attention)
Sync	az webapp deployment source sync --resource-group liveability --name liveability