**ContainedEnvironments Cheat Sheet**

| **Layer / Prompt** | **Where you actually are** | **Typical prompt look** | **What you can do here** | **One-line “jump” to next layer** |
| --- | --- | --- | --- | --- |
| **Windows host – PowerShell** | Your local OS shell | PS C:\MonCode\KonnaxionV14> | • git pull / commit• docker compose up -d• Edit files in VS Code | docker compose exec web bash → into **web container** |
| **Docker – web container** (python:3.13-…) | Isolated Linux shell running Django code | root@abcdef:/app# | • python manage.py makemigrations• python manage.py migrate• Run tests: pytest | exit returns to **PowerShell** |
| **Docker – postgres container** | PostgreSQL service only | (no shell by default) | • psql -U konnaxion -h postgres konnaxion (from *web* container)• Inspect tables, run SQL | \q quits psql |
| **Python virtual-env** | *Inside* the web container (pip list shows project deps) | Same prompt; venv auto-activated | Install/upgrade libs: pip install package==x.y | N/A – exits with exit |
| **Django manage-py commands** | Still inside web container | Same prompt | python manage.py … (collectstatic, shell, etc.) | N/A |
| **Django Admin UI** | Browser at http://localhost:8000/admin/ | (web page) | CRUD for models, view logs | Close tab |
| **DB superuser (psql)** | Postgres CLI inside *web* container or host | konnaxion=# | DDL / manual queries | \q |

**Quick command set**

# 0. Start whole stack

docker compose up -d # web + postgres

# 1. Enter the Django ‘web’ service

docker compose exec web bash # now in /app

# 2. Typical Django ops

python manage.py makemigrations

python manage.py migrate

python manage.py createsuperuser

python manage.py runserver 0.0.0.0:8000

# 3. Optional: open psql (still inside web)

psql -U konnaxion -h postgres konnaxion

# 4. Leave container

exit