## 1. Check Container Names and Ports

powershell

*# List all running containers with names and ports*

docker ps --format "table {{.Names}}\t{{.Image}}\t{{.Ports}}"

*# Alternative format for better readability*

docker ps --format "table {{.Names}}\t{{.Status}}\t{{.Ports}}"

## 2. Get Container IP Address

powershell

*# Get IP address of PostgreSQL container*

docker inspect airflow3x2-postgres-1 --format='{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}'

*# Alternative command to get detailed network info*

docker inspect airflow3x2-postgres-1 | Select-String "IPAddress"

## 3. Check Docker Networks

powershell

*# List all Docker networks*

docker network ls

*# Inspect the specific network your containers are on*

docker network inspect airflow3x2\_default

*# Find which containers are on the network*

docker network inspect airflow3x2\_default --format='{{range .Containers}}{{.Name}} - {{.IPv4Address}}{{"\n"}}{{end}}'

## 4. Test Network Connectivity

powershell

*# Test if PostgreSQL port is accessible from host*

Test-NetConnection -ComputerName localhost -Port 5432

*# Test connection to container IP (replace with actual IP)*

Test-NetConnection -ComputerName 172.18.0.3 -Port 5432

*# Alternative telnet test*

telnet localhost 5432

## 5. Check Container Details

powershell

*# Get full container information*

docker inspect airflow3x2-postgres-1

*# Check if containers are running and healthy*

docker ps -a --filter "name=postgres"

docker ps -a --filter "name=pgadmin"

## 6. Check Docker Compose Services

powershell

*# If you're in the docker-compose directory*

docker-compose ps

*# Check service names from docker-compose file*

Get-Content docker-compose.yaml | Select-String "^\s\*\w+:"

## 7. Find the Correct Host Name

powershell

*# Get all container names in your network*

docker network inspect airflow3x2\_default --format='{{json .Containers}}' | ConvertFrom-Json | ForEach-Object { $\_.PSObject.Properties.Value.Name }

*# Get service aliases*

docker inspect airflow3x2-postgres-1 --format='{{range $key, $value := .NetworkSettings.Networks}}{{range $value.Aliases}}{{.}}{{"\n"}}{{end}}{{end}}'

## Quick Diagnostic Script

powershell

*# Run this complete diagnostic*

Write-Host "=== Container Status ===" -ForegroundColor Green

docker ps --format "table {{.Names}}\t{{.Status}}\t{{.Ports}}"

Write-Host "`n=== PostgreSQL Container IP ===" -ForegroundColor Green

docker inspect airflow3x2-postgres-1 --format='{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}'

Write-Host "`n=== Network Aliases ===" -ForegroundColor Green

docker inspect airflow3x2-postgres-1 --format='{{range $key, $value := .NetworkSettings.Networks}}{{range $value.Aliases}}{{.}}{{"\n"}}{{end}}{{end}}'

Write-Host "`n=== Port Test ===" -ForegroundColor Green

Test-NetConnection -ComputerName localhost -Port 5432 -InformationLevel Quiet

**Run these commands and share the output.** This will help identify the exact container name and IP to use in pgAdmin.

Most likely, you'll find that the correct host name is either:

* postgres
* airflow3x2-postgres-1
* Or the IP address from the network inspect command

A screenshot of a computer

AI-generated content may be incorrect.