Documentation de l'environnement DNS Measures

Mesures DNS dans l'espace et le temps

Projet de mémoire - 2026

20 janvier 2026

Table des matières

1 Introduction

Ce document fournit une documentation complète de l'environnement Docker développé pour le projet de mémoire "Mesures DNS dans l'espace et le temps".

1.1 Objectifs du projet

Le DNS est un service distribué initialement prévu pour associer des noms d'hôtes à leur adresse IP. L'information contenue dans le DNS donne un aperçu de la manière dont Internet est structuré et comment les domaines sont administrés.

Le projet vise à enregistrer une partie de l'information fournie par le système DNS en capturant la diversité des réponses dans le temps et dans l'espace. Ces données pourront ensuite être rendues disponibles à des fins de recherche, notamment pour de la simulation réseau.

1.2 Composants du projet

- **Archivage DNS** : Outil d'archivage des informations DNS
- **Sources de données** : Utilisation de la Tranco list pour les noms de domaines
- **Infrastructure de mesure** : RIPE Atlas pour lancer des requêtes DNS depuis différents lieux
- **Optimisation** : Stratégie pour optimiser les informations archivées
- **Stockage** : Structure de données facilitant le partage des données récoltées

2 Structure du projet

2.1 Organisation des répertoires

```
dns-measures/
  docs/                  # Documentation complète du projet
     documentation.md    # Ce fichier (source Markdown)
     documentation.tex   # Version LaTeX générée
     documentation.pdf   # Version PDF finale
  sources/               # Articles académiques et références
     A High-Performance, Scalable Infrastructure...pdf
     TRANCO A Research-Oriented Top Sites Ranking...pdf
     README.md
  data/                  # Données brutes et traitées
     raw/                # Données brutes (non versionnées)
     processed/          # Données traitées (non versionnées)
  notebooks/             # Notebooks Jupyter pour analyses exploratoires
  scripts/               # Scripts Python pour analyses et traitements
     example_dns_measures.py
     document_utils.py
  reports/               # Rapports générés et analyses finales
  latex/                 # Sources LaTeX du mémoire
     chapters/           # Chapitres du mémoire
     figures/            # Figures et images
     bibliography.bib    # Références bibliographiques
  output/                # Fichiers de sortie (PDF, graphiques)
  .devcontainer/         # Configuration VSCode Dev Container
  .claude/               # Configuration Claude Code
  Dockerfile             # Image Docker principale
  docker-compose.yml     # Orchestration des services
  docker-entrypoint.sh   # Script d'entrée du container
  requirements.txt       # Dépendances Python
  .gitignore             # Fichiers ignorés par Git
  .dockerignore          # Fichiers ignorés par Docker
  readme.md              # Description du sujet de mémoire
  claude.md              # Journal de travail et bonnes pratiques
```

2.2 Rôle de chaque répertoire

2.2.1 `docs/`

Documentation technique complète de l'environnement et des outils. Ce répertoire contient la documentation au format Markdown, LaTeX et PDF.

2.2.2 `sources/`

Articles académiques et références PDF utilisés pour la recherche. Tous les PDFs de référence doivent être placés ici avec une nomenclature claire : `auteur_année_titre_court.pdf`.

2.2.3 `data/`

Données du projet. Les sous-répertoires `raw/` et `processed/` ne sont pas versionnés dans Git mais sont persistés via les volumes Docker.

2.2.4 `notebooks/`

Notebooks Jupyter pour analyses exploratoires et visualisations interactives. Nomenclature recommandée : `YYYY-MM-DD_description.ipynb`.

2.2.5 `scripts/`

Scripts Python de production pour analyses, traitement de données et automatisation.

2.2.6 `reports/`

Rapports d'analyse générés (PDF, HTML, Markdown). Les fichiers générés ne sont pas versionnés.

2.2.7 `latex/`

Sources LaTeX du mémoire final. Organisation en chapitres avec une structure modulaire.

2.2.8 `output/`

Fichiers de sortie générés : graphiques, exports, résultats. Non versionnés.

3 Environnement Docker

3.1 Vue d'ensemble

L'environnement Docker fournit tous les outils nécessaires pour : - Effectuer des mesures DNS actives - Analyser et visualiser les données - Rédiger le mémoire en LaTeX - Manipuler des documents (PDF, Office, images)

3.2 Image de base

- **Base** : `python:3.11-slim-bookworm`
- **Taille approximative** : 5-8 GB (incluant TeXLive complet)
- **Temps de construction** : 20-30 minutes lors de la première construction

3.3 Outils et bibliothèques installés

3.3.1 DNS et Réseau

Outil/Bibliothèque	Description
`dnspython`	Bibliothèque DNS complète pour Python
`ripe.atlas.cousteau`	API Python pour RIPE Atlas
`ripe.atlas.sagan`	Parsing et analyse des résultats RIPE Atlas
`ripe.atlas.tools`	Outils CLI pour RIPE Atlas
`dnsutils`	Outils DNS en ligne de commande (dig, nslookup)
`bind9-host`	Outil host pour requêtes DNS
`whois`	Client WHOIS
`requests`, `httpx`, `aiohttp`	Bibliothèques HTTP
`netaddr`, `tldextract`	Manipulation d'adresses réseau et domaines

3.3.2 Analyse de données

Bibliothèque	Description
`numpy`	Calcul numérique et arrays
`pandas`	Manipulation et analyse de données
`scipy`	Algorithmes scientifiques
`scikit-learn`	Machine learning
`statsmodels`	Modèles statistiques

3.3.3 Visualisation

Bibliothèque	Description
`matplotlib`	Graphiques 2D
`seaborn`	Visualisations statistiques
`plotly`	Graphiques interactifs
`bokeh`	Visualisations web interactives
`altair`	Grammaire de visualisation déclarative
`folium`	Cartes interactives
`geopandas`	Données géospatiales
`cartopy`	Cartographie

3.3.4 Manipulation de documents

3.3.4.1 PDF

Bibliothèque	Description
`PyPDF2`, `pypdf`	Lecture et manipulation de PDF
`pdfplumber`	Extraction de données tabulaires
`pdf2image`	Conversion PDF vers images
`pikepdf`	Manipulation avancée de PDF
`reportlab`	Génération de PDF
`weasyprint`	HTML/CSS vers PDF
`pdfminer.six`	Extraction de texte
`pymupdf` (fitz)	Manipulation rapide de PDF
`camelot-py`	Extraction de tableaux

3.3.4.2 Documents Office

Bibliothèque	Description
`python-docx`	Création et modification de fichiers Word
`python-pptx`	Création et modification de PowerPoint
`openpyxl`	Lecture/écriture Excel (xlsx)
`xlrd`, `xlsxwriter`	Manipulation Excel
`libreoffice`	Suite bureautique complète

3.3.4.3 Images

Bibliothèque	Description
`Pillow`	Traitement d'images
`opencv-python-headless`	Computer vision
`imageio`	Lecture/écriture d'images
`scikit-image`	Traitement d'images scientifique
`imagemagick`	Suite de conversion d'images

3.3.5 LaTeX et documentation

Outil	Description
`texlive-full`	Distribution LaTeX complète
`latexmk`	Compilation automatisée de LaTeX
`biber`, `bibtex`	Gestion de bibliographie
`pylatex`	Génération de LaTeX depuis Python
`pandoc`	Conversion universelle de documents
`markdown`, `mistune`	Traitement Markdown

3.3.6 Jupyter et notebooks

Bibliothèque	Description
`jupyter`	Environnement de notebooks
`jupyterlab`	Interface moderne pour Jupyter

Bibliothèque	Description
`ipywidgets`	Widgets interactifs
`nbconvert`	Conversion de notebooks

3.3.7 Bases de données et stockage

Bibliothèque	Description
`sqlalchemy`	ORM Python
`sqlite-utils`	Utilitaires SQLite
`tinydb`	Base de données NoSQL légère
`h5py`	Format HDF5
`pyarrow`, `fastparquet`	Format Parquet

3.3.8 Utilitaires

Bibliothèque	Description
`python-dateutil`, `pytz`	Gestion des dates et fuseaux horaires
`tqdm`	Barres de progression
`rich`	Formatage terminal enrichi
`click`, `typer`	Création de CLI
`pyyaml`, `toml`	Parsing de fichiers de config
`python-dotenv`	Variables d'environnement
`loguru`	Logging avancé
`beautifulsoup4`, `lxml`	Web scraping

3.3.9 Tests et qualité

Bibliothèque	Description
`pytest`	Framework de tests
`pytest-asyncio`	Tests asynchrones
`black`	Formatage de code
`isort`	Organisation des imports
`flake8`	Linting
`mypy`	Vérification de types

3.3.10 OCR et extraction de texte

Outil	Description
`tesseract-ocr`	Moteur OCR
`pytesseract`	Interface Python pour Tesseract
Langues : français, anglais	Support multilingue

4 Utilisation de l'environnement

4.1 Démarrage avec Docker Compose

4.1.1 Mode interactif

```
# Démarrer le container
docker-compose up -d dns-measures

# Se connecter au container
docker-compose exec dns-measures bash
```

4.1.2 Mode Jupyter Lab

```
# Démarrer Jupyter Lab
docker-compose --profile jupyter up -d

# Accéder à http://localhost:8889
```

4.2 Démarrage avec VSCode Dev Container

1. Ouvrir le projet dans VSCode
2. Appuyer sur `F1` ou `Ctrl+Shift+P`
3. Sélectionner "Dev Containers: Reopen in Container"
4. L'extension Claude Code sera automatiquement installée

4.2.1 Extensions VSCode installées automatiquement

- `ms-python.python` - Support Python
- `ms-python.vscode-pylance` - IntelliSense Python
- `ms-toolsai.jupyter` - Support Jupyter
- `james-yu.latex-workshop` - Support LaTeX
- `redhat.vscode-yaml` - Support YAML
- `esbenp.prettier-vscode` - Formatage de code
- `anthropic.claude-code` - Assistant IA Claude Code

4.3 Commandes utiles

4.3.1 Jupyter Lab

```
# Démarrer Jupyter Lab sur toutes les interfaces
jupyter lab --ip=0.0.0.0 --allow-root --no-browser

# Avec token désactivé (attention : uniquement en dev local)
jupyter lab --ip=0.0.0.0 --allow-root --no-browser \
  --NotebookApp.token='' --NotebookApp.password=''
```

4.3.2 Compilation LaTeX

```
# Méthode recommandée avec latexmk (gère automatiquement les passes)
latexmk -pdf -interaction=nonstopmode document.tex

# Compilation manuelle (2 passes minimum pour table des matières)
pdflatex document.tex
pdflatex document.tex

# Avec bibliographie (3-4 passes nécessaires)
```

```
pdflatex document.tex
biber document  # ou: bibtex document
pdflatex document.tex
pdflatex document.tex
```

4.3.3 Conversion de documents

```
# Markdown vers PDF via LaTeX
pandoc document.md -o document.pdf --pdf-engine=xelatex

# Markdown vers PDF avec template
pandoc document.md -o document.pdf \
  --pdf-engine=xelatex \
  --template=template.tex \
  --toc --number-sections

# HTML vers PDF
pandoc document.html -o document.pdf

# Word vers PDF (avec LibreOffice)
libreoffice --headless --convert-to pdf document.docx
```

4.3.4 Outils DNS

```
# Requête DNS simple
dig example.com

# Requête DNS avec serveur spécifique
dig @8.8.8.8 example.com

# Information WHOIS
whois example.com

# Résolution d'hôte
host example.com
```

5 Configuration RIPE Atlas

5.1 Prérequis

1. Créer un compte sur https://atlas.ripe.net/
2. Obtenir une clé API depuis le dashboard RIPE Atlas
3. Disposer de crédits pour lancer des mesures

5.2 Configuration de la clé API

5.2.1 Dans le container

```
# Variable d'environnement temporaire
export RIPE_ATLAS_API_KEY="votre-cle-api-ici"

# Ou dans un fichier .env
echo "RIPE_ATLAS_API_KEY=votre-cle-api-ici" > .env
```

5.2.2 Dans docker-compose.yml

```
environment:
  - RIPE_ATLAS_API_KEY=${RIPE_ATLAS_API_KEY}
```

Puis créer un fichier `.env` à la racine :

```
RIPE_ATLAS_API_KEY=votre-cle-api-ici
```

5.3 Utilisation de base

```
from ripe.atlas.cousteau import Measurement, Probe, AtlasRequest

# Configuration
api_key = "votre-cle-api"

# Créer une mesure DNS
measurement = Measurement(
    af=4,  # IPv4
    type="dns",
    target="example.com",
    description="Test DNS measurement"
)

# Lancer la mesure
request = AtlasRequest(
    key=api_key,
    measurements=[measurement],
    sources=[Probe(id=1)]
)

result = request.create()
```

6 Bonnes pratiques

6.1 Compilation LaTeX

RÈGLE ABSOLUE : Toujours faire au minimum 2 passes lors de la compilation LaTeX vers PDF.

6.1.1 Pourquoi 2 passes ?

1. **Première passe** : Génération du contenu et collecte des références
2. **Deuxième passe** : Mise à jour de la table des matières, références croisées, numéros de pages

6.1.2 Avec bibliographie

Lorsque vous utilisez BibTeX ou Biber, **3 à 4 passes sont nécessaires** :

```
pdflatex document.tex    # 1ère passe
biber document           # Traitement bibliographie
pdflatex document.tex    # 2ème passe (intégration biblio)
pdflatex document.tex    # 3ème passe (finalisation)
```

6.1.3 Automatisation recommandée

Utiliser `latexmk` qui gère automatiquement toutes les passes nécessaires :

```
latexmk -pdf -interaction=nonstopmode document.tex
```

6.2 Gestion des données

6.2.1 Règles strictes

- **Jamais de fichiers temporaires** dans le repository Git
- Les fichiers `.pyc`, `__pycache__`, `.ipynb_checkpoints` sont ignorés via `.gitignore`
- Les données volumineuses sont stockées dans `data/` (non versionnées)
- Les volumes Docker assurent la persistance des données

6.2.2 Organisation des données

```
data/
  raw/           # Données brutes (lecture seule)
  processed/     # Données traitées (générées)
  cache/         # Données de cache (régénérables)
```

6.3 Analyses et notebooks

6.3.1 Nomenclature des fichiers

- **Notebooks** : `YYYY-MM-DD_description_breve.ipynb`
  - Exemple : `2026-01-20_exploration_tranco.ipynb`
- **Scripts** : Noms descriptifs en snake_case
  - Exemple : `analyze_dns_responses.py`

6.3.2 Contenu des notebooks

- Toujours inclure des cellules Markdown pour expliquer la démarche
- Commenter le code pour les opérations complexes
- Inclure des visualisations pour faciliter la compréhension
- Sauvegarder les résultats importants dans `output/` ou `reports/`

6.3.3 Scripts de production

- Code propre et documenté (docstrings)
- Gestion des erreurs appropriée
- Logging pour tracer l'exécution
- Tests unitaires dans un répertoire `tests/`

6.4 Documentation

6.4.1 Mise à jour de claude.md

Mettre à jour claude.md après chaque session importante :

- Ajouter les actions effectuées dans "Historique des modifications"
- Noter les décisions importantes et leur justification
- Lister les prochaines étapes
- Documenter les problèmes rencontrés et solutions

6.4.2 Documentation du code

- Docstrings pour toutes les fonctions et classes publiques
- Commentaires inline pour la logique complexe
- README.md dans chaque répertoire pour expliquer son contenu

6.5 Commits Git

6.5.1 Format des messages

```
Titre concis (max 50 caractères)

Description détaillée si nécessaire :
- Point 1
- Point 2
- Point 3

Co-Authored-By: Claude Sonnet 4.5 <noreply@anthropic.com>
```

6.5.2 Règles

- Commits atomiques (une fonctionnalité/correction par commit)
- Messages en français pour ce projet
- Ne jamais commiter de fichiers temporaires, secrets, ou données volumineuses
- Vérifier `.gitignore` avant de commiter

6.6 Organisation des sources

6.6.1 Articles PDF

- Placer tous les PDFs dans `sources/`
- Nomenclature : `auteur_année_titre_court.pdf`
- Mettre à jour `sources/README.md` avec une description

6.6.2 Références bibliographiques

- Maintenir un fichier `latex/bibliography.bib` à jour
- Format BibTeX cohérent
- Inclure tous les champs pertinents (auteur, titre, année, DOI, URL)

7 Dépannage

7.1 Problèmes courants

7.1.1 Port Jupyter déjà utilisé

Symptôme : Erreur "Address already in use" lors du démarrage de Jupyter.

Solution : Modifier le port dans `docker-compose.yml` :

```
ports:
  - "8890:8888"  # Changer 8889 en 8890 ou autre port libre
```

7.1.2 Extension Claude Code non installée

Symptôme : L'extension Claude Code n'apparaît pas dans VSCode.

Solutions : 1. Rebuild du container Dev Container : `F1` > "Dev Containers: Rebuild Container" 2. Vérifier `.devcontainer/devcontainer.json` contient bien `"anthropic.claude-code"` 3. Installer manuellement depuis le marketplace VSCode

7.1.3 Erreur LaTeX compilation

Symptôme : Erreur lors de la compilation LaTeX.

Solutions : 1. Vérifier que `texlive-full` est installé dans le container 2. Regarder les logs d'erreur dans le fichier `.log` 3. Utiliser `latexmk` au lieu de `pdflatex` direct 4. S'assurer d'avoir fait au moins 2 passes

7.1.4 Permissions Docker

Symptôme : Erreur de permissions lors de l'accès aux fichiers.

Solution : - Le container tourne en `root` par défaut (voir `devcontainer.json`) - Pour production, créer un utilisateur non-root dans le Dockerfile

7.1.5 Volumes Docker non persistants

Symptôme : Les données disparaissent après arrêt du container.

Solution : - Vérifier les volumes dans `docker-compose.yml` - S'assurer d'utiliser des volumes nommés pour la persistance - Ne pas utiliser `--rm` si on veut conserver les données

7.1.6 Erreur RIPE Atlas API

Symptôme : Erreur d'authentification avec RIPE Atlas.

Solutions : 1. Vérifier que la clé API est correctement définie 2. Vérifier que la clé n'a pas expiré 3. Vérifier les limites de crédits

7.2 Logs et debugging

7.2.1 Logs Docker

```
# Logs du container
docker-compose logs dns-measures

# Suivre les logs en temps réel
docker-compose logs -f dns-measures
```

7.2.2 Logs Python

Utiliser `loguru` pour un logging avancé :

```
from loguru import logger

logger.add("logs/app.log", rotation="500 MB")
logger.info("Message informatif")
logger.error("Message d'erreur")
```

7.2.3 Debug interactif

```
# Avec ipdb
import ipdb; ipdb.set_trace()

# Avec pdb
import pdb; pdb.set_trace()
```

8 Scripts utilitaires

8.1 example_dns_measures.py

Script d'exemple pour effectuer des mesures DNS avec la Tranco list et RIPE Atlas.

8.1.1 Fonctionnalités

- Récupération de la liste Tranco
- Lancement de mesures DNS via RIPE Atlas
- Sauvegarde des résultats

8.1.2 Utilisation

```
from scripts.example_dns_measures import fetch_tranco_list, measure_domains

# Récupérer le top 1000 de Tranco
tranco = fetch_tranco_list(top_n=1000)

# Effectuer des mesures
results = measure_domains(tranco['domain'].tolist()[:10])
```

8.2 document_utils.py

Utilitaires pour manipulation de documents (PDF, Word, LaTeX).

8.2.1 Fonctionnalités

- Extraction de texte depuis PDF
- Conversion Word vers PDF
- Compilation LaTeX
- Fusion de PDFs

8.2.2 Utilisation

```
from scripts.document_utils import (
    pdf_to_text,
    docx_to_pdf,
    compile_latex,
    merge_pdfs
)

# Extraire texte d'un PDF
text = pdf_to_text('article.pdf')

# Convertir Word vers PDF
docx_to_pdf('rapport.docx', 'rapport.pdf')

# Compiler LaTeX
compile_latex('document.tex')

# Fusionner plusieurs PDFs
merge_pdfs(['doc1.pdf', 'doc2.pdf'], 'merged.pdf')
```

9 Références

9.1 Articles principaux

1. **van Rijswijk-Deij, R., Jonker, M., Sperotto, A., & Pras, A. (2016)**. A High-Performance, Scalable Infrastructure for Large-Scale Active DNS Measurements. *IEEE Journal on Selected Areas in Communications*, 34(6), 1877–1888. https://doi.org/10.1109/JSAC.2016.2558918

2. **Le Pochat, V., Van Goethem, T., Tajalizadehkhoob, S., Korczynski, M., & Joosen, W. (2019)**. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. *Proceedings 2019 Network and Distributed System Security Symposium*. https://doi.org/10.14722/ndss.2019.23386

9.2 Documentation en ligne

- **RIPE Atlas Documentation** : https://atlas.ripe.net/docs/
- **Tranco List** : https://tranco-list.eu/
- **dnspython** : https://dnspython.readthedocs.io/
- **Pandas** : https://pandas.pydata.org/docs/
- **Matplotlib** : https://matplotlib.org/stable/contents.html
- **LaTeX** : https://www.latex-project.org/help/documentation/

9.3 Ressources Docker

- **Docker Documentation** : https://docs.docker.com/
- **Docker Compose** : https://docs.docker.com/compose/
- **VSCode Dev Containers** : https://code.visualstudio.com/docs/devcontainers/containers

10 Annexes

10.1 Variables d'environnement

Variable	Description	Valeur par défaut
`PYTHONPATH`	Chemin de recherche Python	`/workspace`
`JUPYTER_ENABLE_LAB`	Active Jupyter Lab	`yes`
`RIPE_ATLAS_API_KEY`	Clé API RIPE Atlas	(à définir)
`PYTHONUNBUFFERED`	Sortie Python non bufférisée	`1`
`LANG`, `LC_ALL`	Locale	`C.UTF-8`

10.2 Ports exposés

Port	Service	Description
8888	Jupyter Lab	Interface principale Jupyter
8889	Jupyter Lab	Service alternatif (docker-compose)
8080	Serveur web	Optionnel, pour visualisations

10.3 Volumes Docker

Volume	Montage	Description
`.`	`/workspace`	Code source du projet
`dns-data`	`/workspace/data`	Données persistantes
`jupyter-config`	`/root/.jupyter`	Configuration Jupyter

10.4 Fichiers de configuration

Fichier	Description
`.devcontainer/devcontainer.json`	Configuration Dev Container
`.claude/settings.local.json`	Configuration Claude Code
`docker-compose.yml`	Orchestration Docker
`Dockerfile`	Définition de l'image
`.gitignore`	Fichiers ignorés par Git
`.dockerignore`	Fichiers exclus du build Docker
`requirements.txt`	Dépendances Python

Documentation générée le 20 janvier 2026