36 — คู่มือตั้งค่าและรัน PN จาก Local ขึ้น AWS (สำหรับผู้ใช้งาน / ผู้ดูแล)

เอกสารนี้เป็น**คู่มือใช้งานจริง** ทีละขั้น: เตรียมเครื่อง local → สร้างเครื่องมือบน AWS ด้วยมือ → ตั้ง `.env` → ติดตั้งบน EC2 → โหลดคลัง → เชื่อม MCP → งานประจำวัน
แพ็กเกจโค้ด: `app/infra/pn-ec2/`
แผนต้นทุน 33 · รายการเครื่องมือ 34 · รายละเอียดเทคนิค deploy 35

สัญลักษณ์ในคู่มือนี้

สัญลักษณ์	ความหมาย
(มือ)	ต้องทำเองบน AWS Console / เบราว์เซอร์ / หน่วยงาน — สคริปต์ทำแทนไม่ได้ทั้งหมด
(สคริปต์)	รันคำสั่งจาก repo หรือ `app/infra/pn-ec2/scripts/`
(ผู้ดูแลระบบ)	Admin / DevOps
(ผู้ใช้แอป)	เจ้าหน้าที่ที่ใช้เว็บร่าง TOR / อัปโหลดคลัง / ถามผ่าน Quick

Region ที่ล็อกในแผน PN: `ap-southeast-1` · ไม่ใช้ Fargate · Embedding = **Cohere Embed 4** (`cohere.embed-v4:0`) · คลัง = **S3 + S3 Vectors**

---

1. ภาพรวมเส้นทาง Local → Cloud

```
[เครื่อง local]
  1) clone repo · เตรียม .env จาก .env.example
  2) (มือ) สร้างเครื่องมือ AWS ตาม checklist
  3) ใส่ชื่อบัคเก็ต / host / รหัสลง .env
  4) (ทางเลือก) สร้าง dataset zip บน local
       ↓
[EC2 บน AWS]
  5) (สคริปต์) bootstrap / docker compose up
  6) (มือ) TLS + DNS + Security Group
  7) sync หรือ upload คลัง → ingest=true
       ↓
[ผู้ใช้]
  8) เข้าเว็บร่าง TOR · Admin อัปโหลดคลัง
  9) (มือ) ต่อ Amazon Quick / Claude Code เข้า MCP
```

สิ่งที่**ระบบทำอัตโนมัติหลังตั้งค่าแล้ว**: chunk ~4096 · Embed 4 · PutVectors · MCP `retrieve`
สิ่งที่**ต้องทำเองเสมอ**: บัญชี AWS, เปิดโมเดล Bedrock, สร้างบัคเก็ต/indexes, EC2+IAM, DNS/TLS, สุ่มรหัสใน `.env` , เชื่อม Quick, อนุมัติงบ

---

2. บทบาทและหน้าที่

บทบาท	ทำอะไร
ผู้ดูแล AWS / DevOps	สร้าง EC2, IAM, S3, S3 Vectors, Bedrock access, Budgets, TLS, `.env` บนเซิร์ฟเวอร์
Admin ในแอป	อัปโหลดเอกสารคลัง (`/pn/kb/upload` + `ingest=true`), ตรวจกลุ่ม RAG, จัดการผู้ใช้
ผู้ใช้ทั่วไป	เข้าเว็บร่าง/ทบทวน TOR, ถามความรู้ผ่านแชทหรือ Amazon Quick (ไม่ต้องแตะ AWS Console)

---

3. สิ่งที่ต้องมีก่อนเริ่ม

3.1 บนเครื่อง local (ผู้ดูแล)

รายการ	หมายเหตุ
Git	clone โปรเจกต์
Docker Desktop / Docker Engine + Compose	ทดสอบ local หรือ build บน EC2
Python 3.11+	สร้าง dataset (`documents/exports/build_pn_aws_dataset.py`)
AWS CLI v2	sync ไฟล์ขึ้น S3, ตรวจสิทธิ์
บัญชี AWS + สิทธิ์สร้าง EC2/S3/IAM/Bedrock	หรือขอจากทีมคลาวด์หน่วยงาน
โดเมนหรือ hostname	ชี้ Elastic IP (สำหรับ HTTPS / MCP)

3.2 บน AWS (ต้องมีสิทธิ์)

- สร้าง/แก้ EC2, Security Group, Elastic IP
- สร้าง S3 bucket + (ถ้ามี) S3 Vectors vector bucket / indexes
- เปิด **Bedrock model access**
- สร้าง IAM Role + instance profile
- (แนะนำ) AWS Budgets

3.3 สำหรับผู้ใช้แอปอย่างเดียว

- URL เว็บที่ Admin แจ้ง (`https://<โดเมน>`)
- บัญชี login ในระบบ
- (ถ้าใช้ Quick) ให้ Admin ตั้ง MCP connector แล้ว — ผู้ใช้แค่เปิด Quick

---

4. Checklist งานที่ต้องทำด้วยมือทั้งหมด (Master)

ทำตามลำดับนี้ครั้งแรก — ติ๊กทีละข้อ

ก. บัญชีและงบ (มือ · ผู้ดูแล)

- [ ] มีบัญชี AWS (หรือ OU แยก PN)
- [ ] เลือก region **Singapore (`ap-southeast-1`)**
- [ ] สร้าง **Budget** เตือนค่าใช้จ่าย EC2 + Bedrock + S3
- [ ] (แนะนำ) เปิด CloudTrail ในบัญชี

ข. Bedrock (มือ · ผู้ดูแล)

- [ ] Console → Amazon Bedrock → **Model access**
- [ ] ขอ/เปิดโมเดลแชททที่หน่วยงานอนุมัติ → จด model id ใส่ `BEDROCK_MODEL_ID`
- [ ] เปิด **`cohere.embed-v4:0`** → `BEDROCK_EMBEDDING_MODEL_ID=cohere.embed-v4:0`
- [ ] รอสถานะ Access granted (อาจใช้เวลา)

ค. S3 อ็อบเจ็กต์คลัง (มือ · ผู้ดูแล)

- [ ] สร้าง bucket เช่น `pn-tor-kb-<หน่วยงาน>` ใน `ap-southeast-1`
- [ ] Block **Public Access** ทั้งหมด
- [ ] เข้ารหัสตามนโยบายหน่วยงาน (SSE-S3 หรือ KMS)
- [ ] จดชื่อใส่ `PN_S3_BUCKET` และ `MINIO_BUCKET` (ชื่อเดียวกัน)

โครงที่จะเกิดขึ้นหลัง sync/upload (ไม่ต้องสร้างโฟลเดอร์ล่วงหน้าก็ได้):

```
s3://{PN_S3_BUCKET}/rags/procurement-th/sources/
s3://{PN_S3_BUCKET}/rags/procurement-th/manifest/
s3://{PN_S3_BUCKET}/rags/procurement-th/chunks/
s3://{PN_S3_BUCKET}/rags/agency-extra/...
```

ง. S3 Vectors (มือ · ผู้ดูแล)

- [ ] สร้าง **vector bucket** แยกจากบัคเก็ตไฟล์ → `PN_S3_VECTOR_BUCKET`
- [ ] สร้าง index มิติ **1024** ชื่อตรง `rag-groups.yaml`:
- [ ] `procurement-th-embed4-v1`
- [ ] `agency-extra-embed4-v1` (ถ้าใช้กลุ่มสอง)
- [ ] ตั้ง `PN_RETRIEVE_BACKEND=s3_vectors` ใน `.env`

> หมายเหตุ: ชื่อบริการ/หน้าจอ S3 Vectors อาจต่างตาม Console รุ่น — ใช้ CLI/ `s3vectors` ตามเอกสาร AWS ปัจจุบันของ region ได้

จ. IAM สำหรับ EC2 (มือ · ผู้ดูแล)

- [ ] สร้าง IAM Role ชนิด **EC2**
- [ ] วางนโยบายจาก `app/infra/pn-ec2/iam/ec2-instance-profile.json`
- [ ] **แทน** `changeme-pn-tor-kb` → ชื่อ `PN_S3_BUCKET` จริง
- [ ] **แทน** `changeme-pn-tor-vectors` → ชื่อ `PN_S3_VECTOR_BUCKET` จริง
- [ ] Attach เป็น **Instance profile** ของเครื่อง EC2
- [ ] บน EC2 ใน `.env`: **เว้นว่าง** `AWS_ACCESS_KEY_ID` / `AWS_SECRET_ACCESS_KEY` (ใช้ role)

ฉ. EC2 + เครือข่าย (มือ · ผู้ดูแล)

- [ ] Launch EC2: Amazon Linux 2023 หรือ Ubuntu 22.04
- [ ] Instance: **t3.small** (เบา) หรือ **t3.medium** (แนะนำถ้ามี webapp+MCP)
- [ ] Disk ≥ **30 GB** gp3
- [ ] Security Group:
- [ ] inbound **443** จาก IP สำนักงาน / VPN เท่านั้น
- [ ] **22** แคบมาก หรือปิด แล้วใช้ **SSM Session Manager**
- [ ] ผูก **Elastic IP**
- [ ] (มือ) สร้าง DNS A record → Elastic IP = ค่า `PN_PUBLIC_HOST`
- [ ] ติดตั้งบนเครื่อง: Docker, Compose plugin, git, awscli (หรือใช้ `bootstrap` หลังมี repo)

ช. ความลับและโดเมนบน `.env` (มือ · ผู้ดูแล)

- [ ] `cp .env.example .env` แล้วแทนทุก `changeme_*`
- [ ] สุ่ม `JWT_SECRET` (≥ 32 ตัว)
- [ ] สุ่ม `QUICK_MCP_AUTH_VALUE` (เก็บไว้ตั้ง Quick)
- [ ] สุ่ม `POSTGRES_PASSWORD`, `REDIS_PASSWORD`, `NEO4J_PASSWORD`
- [ ] `CORS_ORIGINS=https://$PN_PUBLIC_HOST`
- [ ] `COOKIE_SECURE=true` เมื่อใช้ HTTPS
- [ ] **ห้าม commit** ไฟล์ `.env`

ซ. TLS (มือ · ผู้ดูแล)

- [ ] ออกใบรับรอง (เช่น Let's Encrypt / หน่วยงาน)
- [ ] วาง cert ใต้ `app/infra/pn-ec2/nginx/certs/` ตามที่ README ระบุ
- [ ] คัดลอก `nginx/pn.conf.example` → `nginx/pn.conf` แล้วแก้ชื่อโดเมนถ้าจำเป็น
- [ ] เปิด compose profile `with-nginx`

ฌ. Amazon Quick / Claude Code (มือ · ผู้ดูแล หรือเจ้าหน้าที่ IT)

- [ ] Quick → Connectors → MCP
- [ ] URL: `https://$PN_PUBLIC_HOST/mcp`
- [ ] Header ตาม `QUICK_MCP_AUTH_HEADER` / `QUICK_MCP_AUTH_VALUE`
- [ ] ทดสอบ `list_rag_groups` แล้ว `retrieve` พร้อม `rag_group`
- [ ] (Claude Code) ตั้ง MCP remote URL + auth แบบเดียวกัน

ญ. งานประจำของผู้ใช้แอป (มือ · ตามบทบาท)

- [ ] Admin: สร้างผู้ใช้ / มอบสิทธิ์
- [ ] Admin: อัปโหลดเอกสารคลังครั้งแรก (`ingest=true`)
- [ ] ผู้ใช้: เข้าเว็บทดสอบร่าง TOR / ถาม-ตอบ
- [ ] ผู้ใช้ Quick: ถามคำถามจัดซื้อจัดจ้างผ่าน connector

---

5. ขั้นที่ 1 — เตรียมบนเครื่อง Local

5.1 Clone และเปิดแพ็กเกจ deploy (สคริปต์)

```
git clone <URL-repo-ของหน่วยงาน>
cd <repo>/app/infra/pn-ec2
cp .env.example .env
```

Windows (PowerShell):

```
cd <repo>\app\infra\pn-ec2
Copy-Item .env.example .env
notepad .env
```

ยัง**ไม่ต้อง**ใส่ AWS key บน EC2 — บน laptop ถ้าจะ sync S3 จาก local ให้ใส่คีย์ชั่วคราวหรือใช้ `aws sso login` แล้วตั้ง `AWS_PROFILE` นอก Docker

5.2 แมปค่า `.env` กับของที่สร้างบน AWS

สิ่งที่สร้างบน AWS (มือ)	ตัวแปรใน `.env`
Region Singapore	`AWS_REGION`, `BEDROCK_REGION`, `MINIO_REGION`
S3 object bucket	`PN_S3_BUCKET`, `MINIO_BUCKET`
S3 Vectors vector bucket	`PN_S3_VECTOR_BUCKET`
Index ชื่อใน Console	ตรงกับ `rag-groups.yaml` + `PN_S3_VECTOR_INDEX_*`
Elastic IP / DNS	`PN_PUBLIC_HOST`, `CORS_ORIGINS`
Bedrock chat model	`BEDROCK_MODEL_ID`
Embed 4	`BEDROCK_EMBEDDING_MODEL_ID=cohere.embed-v4:0`
Token MCP	`QUICK_MCP_AUTH_VALUE`

ค่าคงที่ที่แนะนำคงไว้:

```
DEPLOYMENT_MODE=cloud
LLM_PROVIDER=bedrock
EMBEDDING_PROVIDER=bedrock
PN_RETRIEVE_BACKEND=s3_vectors
EMBEDDING_DIMENSIONS=1024
PN_CHUNK_TARGET_TOKENS=4096
MINIO_ENDPOINT=s3.ap-southeast-1.amazonaws.com
MINIO_SECURE=true
MINIO_USE_IAM=true
```

คำอธิบายครบทุกหมวด: ดูคอมเมนต์ใน `.env.example`

5.3 (ทางเลือก) สร้างชุดเอกสารคลังบน local (สคริปต์)

```
# จากราก repo
python documents/exports/build_pn_aws_dataset.py --version 0.5.0
```

ได้ `documents/exports/tor-pn-aws-dataset-v0.5.0.zip` — แตกแล้วตั้ง:

```
PN_DATASET_LOCAL_PATH=/path/to/extracted/rag-pdfs
PN_S3_SOURCES_PREFIX=rags/procurement-th/sources/
```

---

6. ขั้นที่ 2 — สร้างเครื่องมือ AWS (รายละเอียดมือ)

ส่วนนี้ขยายจาก checklist ข้อ 4 — โฟกัสหน้าจอ/ลำดับที่คนมักพลาด

6.1 Bedrock Model access

1. เข้า Console region `ap-southeast-1` (หรือตามที่ Bedrock รองรับในบัญชี)
2. Amazon Bedrock → Model access / Model catalog
3. Enable โมเดลแชท + **Cohere Embed 4**
4. ถ้าสถานะ Pending — รอให้เป็น granted ก่อน ingest/ร่าง TOR

6.2 Object bucket vs Vector bucket

ชนิด	ใช้ทำอะไร	ตัวแปร
S3 ปกติ	PDF/DOCX/JSON, manifest, chunk JSON	`PN_S3_BUCKET`
S3 Vectors	เก็บ embedding 1024-d	`PN_S3_VECTOR_BUCKET`

อย่าใส่ชื่อบัคเก็ตเดียวกันโดยเข้าใจผิดว่าเป็นประเภทเดียวกัน — ในแผน PN แยกชัด

6.3 ตรวจ IAM หลังแก้ชื่อบัคเก็ต

เปิดไฟล์ policy แล้วค้นหา `changeme-` — ต้องไม่เหลือก่อน Attach Role

สิทธิ์หลักที่ EC2 ต้องมี:

- `bedrock:InvokeModel` (แชท + embed)
- `s3:Get/Put/List` บน object bucket
- `s3vectors:PutVectors` / `QueryVectors` / ดู index

6.4 Security Group ขั้นต่ำ

Port	จากใคร	หมายเหตุ
443	สำนักงาน/VPN	เว็บ + `/api` + `/mcp`
80	(ชั่วคราว)	เฉพาะตอนขอ Let's Encrypt แล้วปิดได้
22	IP admin เดียว หรือปิด	แนะนำ SSM แทน

อย่าเปิด 5432 / 6379 / 8765 ออก internet

---

7. ขั้นที่ 3 — ติดตั้งและรันบน EC2

7.1 ย้ายโค้ดและ `.env` ขึ้นเครื่อง (มือ + สคริปต์)

ทางเลือก:

1. `git clone` บน EC2 แล้วสร้าง `.env` บนเครื่อง (แนะนำ)
2. หรือ scp/โฟลเดอร์ `pn-ec2` + `.env` จาก local (ระวังไม่ให้ `.env` หลุด)

```
cd <repo>/app/infra/pn-ec2
chmod +x scripts/*.sh
./scripts/bootstrap-ec2.sh
```

หรือ:

```
docker compose -f docker-compose.pn.yml --env-file .env up -d --build
```

เปิด nginx:

```
cp nginx/pn.conf.example nginx/pn.conf
docker compose -f docker-compose.pn.yml --profile with-nginx --env-file .env up -d
```

ตรวจสุขภาพ:

```
./scripts/healthcheck.sh
```

อัปเดตรอบหลัง:

```
git pull
docker compose -f docker-compose.pn.yml --env-file .env up -d --build
```

7.2 บริการที่ควรเห็นหลังขึ้น

บริการ	บทบาทที่ผู้ใช้สัมผัส
frontend	เว็บ UI
backend	API รวม `/api/v1/pn/kb/*`
mcp-rag	ค้น S3 Vectors
amazon-quick	จุด MCP ที่ Quick ต่อ (`/mcp`)
postgres/redis/mongo	state บน disk EC2 (ไม่ใช่คลังเวกเตอร์)

---

8. ขั้นที่ 4 — โหลดคลังความรู้และ ingest

8.1 Sync กองไฟล์จาก local (สคริปต์ • ผู้ดูแล)

ต้องมี AWS CLI ที่ login แล้ว และ `.env` ชี้ `PN_DATASET_LOCAL_PATH`:

```
# Linux/macOS / Git Bash บน EC2 หรือ laptop
./scripts/sync-dataset-to-s3.sh
```

```
# Windows
.\scripts\sync-dataset-to-s3.ps1
```

หลัง sync มีแค่ไฟล์บน S3 — **ยังไม่มีเวกเตอร์** จนกว่าจะเรียก ingest

8.2 อัปโหลดทีละไฟล์ผ่าน API (ง่ายสุด) (ผู้ดูแล / Admin แอป)

1. Login เว็บด้วยบัญชี **Admin**
2. ได้ JWT (จาก UI หรือ API login)
3. อัปโหลดพร้อมฝัง:

```
curl -X POST "https://$PN_PUBLIC_HOST/api/v1/pn/kb/upload" \
  -H "Authorization: Bearer $TOKEN" \
  -F "file=@ระเบียบ.pdf" \
  -F "rag_group=procurement-th" \
  -F "ingest=true"
```

Pipeline ที่ระบบทำ: เก็บ S3 → ตรวจไฟล์ → ดึงข้อความ → chunk ~4096 → Embed 4 → PutVectors

ฝังไฟล์ที่มีบน S3 แล้ว:

```
curl -X POST "https://$PN_PUBLIC_HOST/api/v1/pn/kb/ingest" \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d "{\"object_key\":\"rags/procurement-th/sources/...\",\"rag_group\":\"procurement-th\"}"
```

ทดสอบค้น (ก่อนต่อ Quick ก็ได้):

```
curl -X POST "https://$PN_PUBLIC_HOST/api/v1/pn/kb/search" \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d "{\"query\":\"ราคากลางที่ปรึกษา\",\"rag_group\":\"procurement-th\",\"top_k\":3}"
```

8.3 เลือก `rag_group`

id	ใช้เมื่อ
`procurement-th`	คลังหลักกฎหมาย/ระเบียบจัดซื้อจัดจ้าง
`agency-extra`	เอกสารเฉพาะหน่วยงาน (แยก index)

ดูรายการ: `GET /api/v1/pn/kb/groups` หรือ MCP `list_rag_groups`

---

9. ขั้นที่ 5 — เชื่อม MCP ให้ผู้ใช้ปลายทาง

9.1 Amazon Quick (มือ)

1. เปิด Quick Enterprise → Connectors
2. เพิ่ม **MCP**
3. Endpoint: `https://$PN_PUBLIC_HOST/mcp`
4. Auth: header ชื่อตาม `QUICK_MCP_AUTH_HEADER` (ปกติ `Authorization`) ค่า = `QUICK_MCP_AUTH_VALUE`
5. ลอง tool:
6. `list_rag_groups`
7. `retrieve` โดยใส่ `query` + `rag_group` (เช่น `procurement-th`) · `top_k` แนะนำ 3

รายละเอียดเพิ่ม: 32-AMAZON-QUICK.md

9.2 Claude Code / MCP client อื่น (มือ)

ตั้ง remote MCP URL เดียวกัน + auth token เดียวกัน
ใช้เฉพาะค้น (`retrieve`) — **ไม่** ingest ผ่าน MCP

---

10. คู่มืองานประจำวัน (ผู้ใช้แอป)

10.1 ผู้ใช้ทั่วไป

งาน	วิธี
ร่าง TOR	เข้า `https://$PN_PUBLIC_HOST` → เวิร์กโฟลว์ร่างตามเมนู
ถามระเบียบ	แชทในเว็บ หรือ Amazon Quick ที่ IT ต่อ MCP แล้ว
อัปโหลดคลังเอง	ทำไม่ได้ถ้าไม่มีสิทธิ์ Admin — ส่งไฟล์ให้ Admin

ไม่ต้องเข้า AWS Console · ไม่ต้องแก้ `.env`

10.2 Admin แอป

งาน	วิธี
เพิ่มเอกสารคลัง	Upload + `ingest=true` หรือ sync แล้ว `/ingest`
ตรวจว่าขึ้นคลัง	`GET /pn/kb/objects?rag_group=...` หรือดูใน S3 Console
ทดสอบค้น	`/pn/kb/search` หรือ MCP
หมุน token MCP	เปลี่ยน `QUICK_MCP_AUTH_VALUE` → restart compose → อัปเดต Quick

10.3 ผู้ดูแลระบบ (รายเดือน/เมื่อมีปัญหา)

งาน	วิธี
ดูงบ	AWS Budgets / Cost Explorer
อัปเดตแอป	`git pull` + compose `--build`
ใบรับรองหมดอายุ	ต่ออายุ TLS แล้วรีสตาร์ท nginx
ดิสก์เต็ม	ตรวจ volume Docker / log
โมเดล Bedrock เปลี่ยน	แก้ `BEDROCK_MODEL_ID` แล้ว restart backend

---

11. เกณฑ์ผ่านก่อนเปิดให้ผู้ใช้จริง

- [ ] Checklist ข้อ 4 (ก–ฌ) ครบ
- [ ] `healthcheck` ผ่าน · เปิดเว็บด้วย HTTPS ได้
- [ ] อัปโหลดทดสอบ 1 ไฟล์ `ingest=true` สำเร็จ
- [ ] `/pn/kb/search` หรือ MCP `retrieve` คืนข้อความภาษาไทยที่เกี่ยวข้อง
- [ ] Quick/Claude เรียก MCP ได้ภายใน ~60 วินาที
- [ ] ผู้ใช้ทั่วไป login และเปิดหน้าแรกได้
- [ ] ไม่มี `changeme_` ค้างในค่าสำคัญบนเซิร์ฟเวอร์
- [ ] Security Group ไม่เปิด DB/Redis สู่ internet

---

12. แก้ปัญหาเบื้องต้น

อาการ	ตรวจอะไร
Upload ล้มเหลว	IAM ถึง object bucket หรือยัง · `MINIO_*` · Admin role
Ingest ล้มเหลว	Bedrock Embed 4 access · `PN_S3_VECTOR_BUCKET` · ชื่อ index ตรงหรือไม่ · มิติ 1024
MCP ว่าง / ไม่มี chunk	ingest แล้วหรือยัง · `rag_group` ถูกหรือไม่ · `PN_RETRIEVE_BACKEND`
Quick 401	`QUICK_MCP_AUTH_VALUE` ไม่ตรง · header ชื่อผิด
เว็บ CORS error	`CORS_ORIGINS` ต้องเป็น `https://` + host จริง
EC2 เรียก Bedrock ไม่ได้	instance profile · region · model access

---

13. สรุปสั้น: อะไรมือ / อะไรอัตโนมัติ

(มือ) ผู้ใช้/ผู้ดูแลทำ	ระบบทำหลังตั้งค่า
สร้าง AWS resources, IAM, SG, DNS, TLS	Docker services บน EC2
เปิด Bedrock models, Budgets	Chunk ~4096 + Embed 4
กรอก `.env`, สุ่มรหัส	Put/Query S3 Vectors

(มือ) ผู้ใช้/ผู้ดูแลทำ	ระบบทำหลังตั้งค่า
Sync/upload เอกสาร, กด ingest	MCP `retrieve` / `list_rag_groups`
ต่อ Quick/Claude, สร้างบัญชีผู้ใช้	ร่าง TOR ผ่าน Bedrock chat
ดูแลงบ, ต่ออายุใบรับรอง, อัปเดตแอป	เก็บ state ใน Postgres/Redis บน volume

---

14. อ้างอิงไฟล์

ไฟล์	ใช้เมื่อ
`app/infra/pn-ec2/.env.example`	แม่แบบคอนฟิก
`app/infra/pn-ec2/README.md`	สรุปในโฟลเดอร์ deploy
`app/infra/pn-ec2/rag-groups.yaml`	ชื่อกลุ่ม + vector index
`app/infra/pn-ec2/iam/ec2-instance-profile.json`	สิทธิ์ EC2
`app/infra/pn-ec2/scripts/`	bootstrap, sync, healthcheck
35	รายละเอียดเทคนิค deploy ย่อ
34	รายการเครื่องมือ AWS
33	แผนต้นทุนและสถาปัตยกรรม
32	Amazon Quick ละเอียด

สร้าง PDF: `python Discussions/_export_pn_plan_pdf.py` (รวมไฟล์ 36)