Secure AI Knowledge Workbench

1. What this product is

Secure AI Knowledge Workbench is a self-hosted web + API service that lets you upload documents into your own AWS account and then ask questions that are answered using Amazon Bedrock models grounded in your uploaded content. It automates the "upload → ingest → ask" workflow and provides a secure interface on an EC2 instance.

2. Key concepts you should understand

- Public IP: You will access the Workbench using the EC2 instance's public IP address.
- Setup Code: A unique secret generated on first boot and stored on the instance. It is the Basic Auth password for /setup and all /api/* endpoints.
- Basic Auth username: always admin.
- Your documents: Either upload from your computer (you must know the local file path) or place files directly into the Workbench's S3 bucket prefix and then trigger ingestion.
- AWS resources: The Workbench uses your AWS account to store documents in S3 and build a Bedrock Knowledge Base backed by S3 Vectors (and/or related vector resources).

3. Requirements

- An AWS account with permission to use Amazon Bedrock in your region.
- EC2 security group allowing inbound TCP 443 (and optionally 80) from your IP.
- An IAM role attached to the instance with permissions for: S3, S3 Vectors, and Bedrock Knowledge Bases (and iam:PassRole for Bedrock when creating the KB role).
- A modern web browser (Chrome/Edge/Firefox) for the web UI, or curl for API testing.

4. Launch the instance

1) Launch the product via 1-click from AWS Marketplace. **Wait** until the instance status changes to 'Running' and passes all health checks. Then, connect to your instance using your Amazon private key and the '**ubuntu**' user."

2) In the EC2 console, select your new instance and note the instance's Public IPv4 address (this is your <EC2_PUBLIC_IP>).

3) Ensure your security group allows inbound HTTPS (TCP 443) from your IP address. If you cannot reach the Workbench, this is the #1 thing to check.

5. Attach the required IAM role to the instance

The Workbench needs an IAM Role (Instance Profile) attached to your EC2 instance so it can securely call AWS services (Bedrock Knowledge Bases, S3, S3 Vectors) without storing AWS access keys on the server.

Attach the role in the AWS Console:

1. Go to AWS Console → EC2 → Instances.
2. Select your Secure AI Knowledge Workbench instance.
3. Click Actions → Security → Modify IAM role.
4. Under IAM role, select the role you created (example: SecureAIKnowledgeWorkbenchRole).
5. Click Update IAM role.

Confirm on the server:

SSH into the instance and run:

```
aws sts get-caller-identity
```

You should see an assumed-role ARN for the instance role.

6. Start the Workbench containers after launch (required)

After the instance boots, you must start the Workbench application stack (Docker containers). SSH into the instance and run:

```
cd /opt/secure-rag-workbench
sudo docker compose up -d
sudo docker ps
```

You should see three containers running: nginx, api, and redis. If you do not start the containers, the HTTPS endpoints will not respond.

7. Find your Setup Code and open /setup (first run)

Retrieve the Setup Code (password for Basic Auth):

```
sudo cat /var/lib/secure-rag-workbench/setup_code
```

Open the Setup page in your browser (required once per new instance):

```
https://<EC2_PUBLIC_IP>/setup
```

What /setup is for:
• It completes first-run initialization for this instance.
• It verifies/records the AWS resource configuration (Knowledge Base IDs, S3 bucket/prefix, vector index references) so the API knows what to use.
• It confirms your Basic Auth is working.

Login when prompted:
• Username: admin
• Password: <SETUP_CODE>

8. Verify the Workbench is running (health check)

Run this from any machine with curl installed:

```
curl -k -u "admin:<SETUP_CODE>" https://<EC2_PUBLIC_IP>/api/health
```

Expected response is JSON like:

```
{"ok":true,"region":"us-east-1","knowledgeBaseId":"...","dataSourceId":"...","dataBucket":"..."}
```

9. Upload documents for ingestion

You have two ways to add documents. The recommended method is using /api/ingest because it both uploads the file to the correct S3 prefix and triggers a Bedrock ingestion job automatically.

9.1 Recommended: Upload from your computer using the API

You must know the path to the file on your computer. Examples:

Linux/macOS example:

```
curl -k -u "admin:<SETUP_CODE>" \
  -F "file=@/path/to/YourDocument.pdf" \
  https://<EC2_PUBLIC_IP>/api/ingest
```

Windows PowerShell example (must use curl.exe):

```
$IP="3.95.16.21"
$CODE="<SETUP_CODE>"
curl.exe -k -u "admin:$CODE" `
  -F "file=@C:\Path\To\YourDocument.pdf" `
  "https://$IP/api/ingest"
```

Expected response: JSON showing the uploaded S3 bucket/key and an ingestionJob object that starts processing your document.

9.2 Advanced: Upload directly to S3 (then start ingestion)

Some users prefer uploading directly to S3 (e.g., from an existing pipeline). If you upload directly to S3, you must still trigger an ingestion job to index the new files.

1) Find the Workbench data bucket from /api/health output (dataBucket).

2) Upload your file into the docs/ prefix, for example:

```
aws s3 cp "YourDocument.pdf" s3://<DATA_BUCKET>/docs/
```

3) Trigger ingestion using the Knowledge Base and Data Source IDs stored on the instance.

View resource IDs:

```
sudo cat /var/lib/secure-rag-workbench/aws_resources.json
```

Start ingestion (example):

```
aws bedrock-agent start-ingestion-job \
  --knowledge-base-id <KB_ID> \
  --data-source-id <DS_ID> \
  --region <AWS_REGION>
```

10. Ask questions (examples + what to expect)

Once ingestion completes, you can ask questions and receive answers grounded in your uploaded documents. This is the core purpose of the Workbench.

Ask via API:

```
curl -k -u "admin:<SETUP_CODE>" \
  -H "Content-Type: application/json" \
  -d '{"text":"Summarize the uploaded document in one sentence and quote one key phrase."}' \
  https://<EC2_PUBLIC_IP>/api/ask
```

Expected response:

```
{"answer":"..."}
```

Example questions:

- “Summarize the document in one sentence.” → You get a concise summary grounded in the doc.
- “List the key points as bullets.” → You get bullet points extracted from the doc’s content.
- “What does the document say about X?” → You get a direct answer based on relevant passages.
- “Quote one sentence that supports your answer.” → You get a short quote when available.

Important: If you ask questions before ingestion finishes, answers may be incomplete. Wait a minute and try again.

11. What happens to your files (data handling)

• Your uploaded documents are stored in an S3 bucket in your AWS account (not in the vendor’s account).
• The Workbench places documents under the docs/ prefix and triggers a Bedrock ingestion job.
• Bedrock Knowledge Bases chunk and embed your content using an embedding model (e.g., Titan Embed v2) and store vectors in your configured vector index.
• Your security controls apply: S3 public access is blocked and bucket encryption is enabled.
• You control lifecycle and deletion: deleting S3 objects removes the source files; re-ingestion updates the index.

12. Troubleshooting (common issues)

- Cannot connect to https://<EC2_PUBLIC_IP>: Check the instance security group allows inbound TCP 443 from your IP. Also confirm containers are running.
- Health endpoint returns 401: Username must be admin and password must be the Setup Code from /var/lib/secure-rag-workbench/setup_code.
- Health endpoint returns 502/500: Restart containers (sudo docker compose restart). Check logs: sudo docker compose logs --tail=200 nginx and api.
- Ingestion job says CREATING/STARTING: Wait and retry; Knowledge Base creation must reach ACTIVE before ingestion can run.
- Bedrock model access denied: In your AWS account, enable access to the selected embedding model and at least one generation model in the Bedrock console.

Appendix A: IAM permissions overview (high level)

The attached instance role must allow access to:
• S3 (read/write to the Workbench data bucket)
• S3 Vectors (create/list/get indexes and vector buckets used by the Knowledge Base)
• Bedrock Agent / Knowledge Bases APIs (create KB, data source, start ingestion)

• iam:PassRole to allow Bedrock to assume the KB service role when creating the Knowledge Base

Your organization may require tighter scoping. Use least-privilege in production.