Secure AI Knowledge Workbench

AWS Marketplace Quick Start and Customer Guide

What this does: Launch a private document intelligence workspace in your AWS account. Upload documents, wait for indexing, then ask grounded questions from one protected browser dashboard.

Start here

Before launch	Requirement
AWS Region	US East (N. Virginia) us-east-1
Instance	t3.large for evaluation; t3.xlarge for regular business use
Inbound access	TCP 443 for the browser; TCP 22 for administrator SSH. Restrict both to trusted IPs.
IAM role	Create and attach your own EC2 role before opening /setup. Use the companion policy JSON file.

Important: All buckets, vector storage, Knowledge Base resources, and AWS charges remain in your AWS account. Do not add AWS access keys to the instance.

The companion policy file is at the end of the document: Secure-AI-Knowledge-Workbench-v2.3-Customer-IAM-Policy.json

1. Create and attach your EC2 IAM role

Why this is required: The Workbench uses your EC2 IAM role to create and operate the private Amazon Bedrock, Amazon S3, and S3 Vectors resources in your account. No long-term AWS keys are stored on the server.

1. Open IAM > Policies > Create policy, select the JSON tab, and paste the contents of the companion JSON file.
2. Replace every <YOUR_AWS_ACCOUNT_ID> placeholder with your own 12-digit AWS account ID. Name the policy SecureAIKnowledgeWorkbenchProvisionerPolicy.
3. Open IAM > Roles > Create role. Select AWS service, then EC2. Attach SecureAIKnowledgeWorkbenchProvisionerPolicy and name the role SecureAIKnowledgeWorkbenchEC2Role.
4. Attach the role to the instance: EC2 > select instance > Actions > Security > Modify IAM role.

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

Continue only when: The command returns an assumed-role ARN for the instance.

2. Launch and connect

Security group port	Use
22 / TCP	SSH administration. Limit to trusted administrator IPs or CIDR ranges.
443 / TCP	Secure setup page and Workbench dashboard. Allow trusted browser users.
80 / TCP	Optional HTTP-to-HTTPS redirect.
8000 and 6379	Internal services. Do not expose publicly.

Launch the Marketplace AMI in us-east-1, select your EC2 key pair, and wait until EC2 status checks pass. Use the default ubuntu username:

```
ssh -i /path/to/your-key.pem ubuntu@YOUR_INSTANCE_PUBLIC_IP
```

3. Wait for first boot

First boot generates a fresh certificate and one-time setup code, then starts the Workbench. Do not run Docker Compose manually.

```
codecreator-secure-rag-status
cat /home/ubuntu/FIRST_LOGIN.txt
```

Open the exact HTTPS setup address shown in FIRST_LOGIN.txt.

Browser certificate notice: The default public-IP setup uses a self-signed HTTPS certificate. Confirm the public IP in the browser matches your EC2 instance before continuing. Use your organization certificate process when you later use a production domain.

4. Complete first-run setup

1. Retrieve the one-time Setup Code.

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

2. Open the /setup address from FIRST_LOGIN.txt. Paste the Setup Code, enter a workspace name, create an admin password of at least 14 characters, and select Create My Private AI Workspace.

3. Stay on the automatically refreshing setup status page. Do not sign in yet.

Normal in-progress message: Status: running | Step: ingestion | Starting the initial document sync.

4. Proceed only when the page says: “Secure AI Knowledge Workbench is ready. Your private AWS workspace was created **successfully**.” Then open the Workbench link and sign in as admin.

5. Upload, index, and ask

Sign in with username admin and the password created during setup. The Setup Code is not your dashboard password.

1. In Upload a document, select your file. Microsoft Word DOCX was validated in the v2.3 buyer test.

2. Select Upload and index. Wait before asking questions.

Required readiness message: Indexing complete. Your document is ready for questions.

3. In Ask your knowledge base, enter a question about the indexed content and select Ask securely.

Grounded-answer behavior: When the document does not contain the answer, the Workbench indicates it could not find the information in the uploaded documents.

6. Everyday operation and support

Command	Purpose
codecreator-secure-rag-status	Shows first-boot, service, container, and local-health status.
codecreator-secure-rag-url	Shows the current public setup or Workbench address.
sudo cat /var/lib/secure-rag-workbench/aws_resources.json	Shows resource IDs created for this workspace.

When you are done: Terminating the EC2 instance does not automatically delete the S3 bucket, S3 Vectors resources, Knowledge Base, data source, or related service role. Use aws_resources.json and your organization’s change process to remove them.

Troubleshooting and resource cleanup

Issue	Check
Setup page does not open	Confirm inbound TCP 443, wait for first boot, then run codecreator-secure-rag-status.
Setup remains on status	Confirm the EC2 role is attached with aws sts get-caller-identity, then allow the page to refresh.
Dashboard sign-in fails	Use admin and the password created during setup, not the one-time Setup Code.
New document is not answered	Confirm Upload and index was selected and wait for the required indexing-complete message.

When you are done: Terminating the EC2 instance does not automatically delete the S3 bucket, S3 Vectors resources, Knowledge Base, data source, or related service role. Use aws_resources.json and your organization's change process to remove them.

Official AWS resources

- Connect to a Linux EC2 instance
- Configure EC2 security groups
- Attach an IAM role to an EC2 instance
- Amazon Bedrock Knowledge Bases
- Knowledge Base permissions
- Amazon S3 Vectors with Knowledge Bases
- Knowledge Base data requirements

IAM role reminder: Use the companion policy JSON file below to create the permissions policy, then create an EC2 role in the IAM console and attach it to the instance before opening /setup.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "BedrockKnowledgeBaseProvisioning",
      "Effect": "Allow",
      "Action": [
        "bedrock:CreateKnowledgeBase",
        "bedrock:CreateDataSource",
        "bedrock:GetKnowledgeBase",
        "bedrock:GetDataSource",
        "bedrock:ListKnowledgeBases",
        "bedrock:ListDataSources",
        "bedrock:StartIngestionJob",
        "bedrock:GetIngestionJob",
        "bedrock:ListIngestionJobs",
        "bedrock:TagResource"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestedRegion": "us-east-1"
        }
      }
    },
    {
      "Sid": "BedrockGroundedAnswers",
      "Effect": "Allow",
      "Action": [
        "bedrock:Retrieve",
        "bedrock:RetrieveAndGenerate"
      ],
      "Resource": "arn:aws:bedrock:us-east-1:<YOUR_AWS_ACCOUNT_ID>:knowledge-base/*"
    },
    {
      "Sid": "InvokeConfiguredGenerationModel",
      "Effect": "Allow",
      "Action": "bedrock:InvokeModel",
      "Resource": "arn:aws:bedrock:us-east-1::foundation-model/amazon.nova-lite-v1:0"
    },
    {
      "Sid": "DescribeConfiguredModel",
      "Effect": "Allow",
      "Action": "bedrock:GetFoundationModel",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestedRegion": "us-east-1"
        }
      }
    },
    {
      "Sid": "CreateSecureDocumentBucket",
      "Effect": "Allow",
      "Action": "s3:CreateBucket",
      "Resource": "arn:aws:s3:::secure-ai-kb-<YOUR_AWS_ACCOUNT_ID>-*"
    },
    {
      "Sid": "ConfigureSecureDocumentBucket",
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:PutBucketOwnershipControls",
        "s3:PutBucketPublicAccessBlock",
        "s3:PutBucketTagging",
        "s3:PutEncryptionConfiguration"
      ],
      "Resource": "arn:aws:s3:::secure-ai-kb-<YOUR_AWS_ACCOUNT_ID>-*"
```

```
    },
    {
      "Sid": "UseSecureDocumentObjects",
      "Effect": "Allow",
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:GetObject",
        "s3:ListMultipartUploadParts",
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::secure-ai-kb-<YOUR_AWS_ACCOUNT_ID>-*/*"
    },
    {
      "Sid": "CreateAndManageWorkbenchVectorBuckets",
      "Effect": "Allow",
      "Action": [
        "s3vectors:CreateVectorBucket",
        "s3vectors:GetVectorBucket",
        "s3vectors:GetVectorBucketPolicy",
        "s3vectors:ListIndexes",
        "s3vectors:PutVectorBucketPolicy",
        "s3vectors:TagResource"
      ],
      "Resource": "arn:aws:s3vectors:us-east-1:<YOUR_AWS_ACCOUNT_ID>:bucket/secure-ai-vectors-<YOUR_AWS_ACCOUNT_ID>-*"
    },
    {
      "Sid": "CreateAndManageWorkbenchVectorIndexes",
      "Effect": "Allow",
      "Action": [
        "s3vectors:CreateIndex",
        "s3vectors:GetIndex",
        "s3vectors:TagResource"
      ],
      "Resource": "arn:aws:s3vectors:us-east-1:<YOUR_AWS_ACCOUNT_ID>:bucket/secure-ai-vectors-<YOUR_AWS_ACCOUNT_ID>-
*/index/documents-*"
    },
    {
      "Sid": "CreateBedrockKnowledgeBaseServiceRole",
      "Effect": "Allow",
      "Action": [
        "iam:CreateRole",
        "iam:GetRole",
        "iam:PutRolePolicy",
        "iam:TagRole",
        "iam:UpdateAssumeRolePolicy"
      ],
      "Resource": "arn:aws:iam::<YOUR_AWS_ACCOUNT_ID>:role/SecureAIKnowledgeWorkbenchKB-*"
    },
    {
      "Sid": "PassOnlyWorkbenchRoleToBedrock",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::<YOUR_AWS_ACCOUNT_ID>:role/SecureAIKnowledgeWorkbenchKB-*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "bedrock.amazonaws.com"
        }
      }
    }
  ]
}
```