

RoostGPT CLI Installation Guide - EC2 Linux

Overview

RoostGPT is an AI-powered testing tool. This guide will help you install RoostGPT on your EC2 (RHEL or Ubuntu) instance in just a few minutes.

Pilot Prerequisites

Before installing RoostGPT, ensure your system meets the following requirements:

Requirement	Specification
Operating System	RHEL 9.5 or Ubuntu 22.04
CPU (or vCPUs)	8 or more
RAM	16 GB or more
Instance Type	c5a.2xlarge or better
Root Disk Space	Minimum 100 GB available
Additional Volume	Minimum 150 GB
Mount Point for additional volume	/var/tmp/Roost (with full permissions granted to SSH user)
Network	Internet connection enabled
Permissions	sudo/root access required (SSH user should have sudo permissions)
AWS Key & Secret	Set in environment or in .profile for accessing AWS Services

Requirement	Specification
AWS IAM User Role	Policy: AmazonBedrockFullAccess or Custom Bedrock Policy as defined below
AWS Bedrock Inference Profiles (one or more of these listed profile id)	global.anthropic.claude-sonnet-4-5-20250929-v1:0 global.anthropic.claude-sonnet-4-20250514-v1:0 eu.anthropic.claude-sonnet-4-5-20250929-v1:0 eu.anthropic.claude-sonnet-4-20250514-v1:0 eu.anthropic.claude-3-7-sonnet-20250219-v1:0
RoostGPT License & Installer	Provided by Roost Team
POC 1 - Input Sample	1. Word document or PDF with User Story or System Analysis Document 2. Optional: API Specification that is representative of current features
POC 2 - Input Sample	1. Swagger API Spec

AWS Bedrock Model Access to one or more of the anthropic (cross-region) models is fine.

Custom Bedrock Policy for AWS IAM User

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "BedrockAPIs",
      "Effect": "Allow",
      "Action": [
        "bedrock:Get*",
        "bedrock:List*",
        "bedrock:CallWithBearerToken",
        "bedrock:BatchDeleteEvaluationJob",
        "bedrock:CreateEvaluationJob",
        "bedrock:CreateGuardrail",
        "bedrock:CreateGuardrailVersion",
        "bedrock:CreateInferenceProfile",
        "bedrock:CreateModelCopyJob",
        "bedrock:CreateModelCustomizationJob",
```

```

    "bedrock:CreateModelImportJob",
    "bedrock:CreateModelInvocationJob",
    "bedrock:CreatePromptRouter",
    "bedrock:CreateProvisionedModelThroughput",
    "bedrock>DeleteCustomModel",
    "bedrock>DeleteGuardrail",
    "bedrock:DeleteImportedModel",
    "bedrock:DeleteInferenceProfile",
    "bedrock>DeletePromptRouter",
    "bedrock>DeleteProvisionedModelThroughput",
    "bedrock:StopEvaluationJob",
    "bedrock:StopModelCustomizationJob",
    "bedrock:StopModelInvocationJob",
    "bedrock:TagResource",
    "bedrock:UntagResource",
    "bedrock:UpdateGuardrail",
    "bedrock:UpdateProvisionedModelThroughput",
    "bedrock:ApplyGuardrail",
    "bedrock:InvokeModel",
    "bedrock:InvokeModelWithResponseStream"
  ],
  "Resource": "*"
}
]
}

```

- Configure AWS profile into launched server, either by
 - setting **AWS_ACCESS_KEY_ID** and **AWS_SECRET_ACCESS_KEY**, in \$HOME/.profile
 - Or stage \$HOME/.aws/credentials
- Transfer roostgpt-installer executable into the EC2 instance
- Ensure the below is installed on the instance to support troubleshooting.
 - Python3 (version 3.3 or later)
 - AWS CLI (version aws-cli/2.31.29 or later)

Details of Roost Installer

Bundled artifact in the Roost installer

Name	Purpose / Description	Location
roostgpt-linux	CLI executable for roostGPT	Bundled binary Reference: v1.1.18/roostgpt-linux
roostenc-linux	Decrypt/Encrypt binary for troubleshooting RoostGPT behaviour	Bundled in the roostgpt-installer (not available publicly)
license/license.ral	Roost License File	Bundled in the roostgpt-installer (not available publicly)
workspace/postman-api-test	Sample swagger API spec file and env to generate postman collection with tests	API Spec: weather,yaml Reference: https://openweathermap.org/api
workspace/functional-test	Sample Functional Test project with env	Sample word document written by Roost Team - (not publicly available)

Installation Steps

Mount EBS Volume (optional, check if it is done already)

1. Verify that /var/tmp/Roost does not exist already
2. Check the read/write/execute permissions on the mount point
3. Extract the volume ID
4. Mount the disk, volume as Roost folder
5. Add fstab entry for handling m/c reboot

```
# Verify if the Roost folder exists
df -h /var/tmp/Roost

# if not create directory
sudo mkdir $ROOST_DIR
sudo chown `id -u`:`id -g` ${ROOST_DIR}

# Check if volume and disk exists
# Typically this is the
export DISK="nvme0n1"
export EBS_VOLUME="/dev/nvme0n1"
blkid | grep "${DISK}:"
```

```
lsblk | grep -w "${EBS_VOLUME}"
```

```
# Run below snippet to mount disk and add entry to fstab file
```

```
lsblk -f | grep -w "${DISK}" | grep "ext4"
```

```
if [ $? -ne 0 ]; then
```

```
    sudo mkfs -t ext4 ${EBS_VOLUME}
```

```
fi
```

```
if [ ! -d "${ROOST_DIR}" ]; then
```

```
    sudo mkdir ${ROOST_DIR}
```

```
fi
```

```
sudo mount ${EBS_VOLUME} ${ROOST_DIR}
```

```
mount | grep "$EBS_VOLUME"
```

```
if [ $? -eq 0 ]; then
```

```
    sudo chown `id -u`:`id -g` ${ROOST_DIR}
```

```
grep -v "${ROOST_DIR}" /etc/fstab | sudo tee /etc/fstab.noroost
```

```
# Plan for reboot
```

```
epoch=$(date +%s)
```

```
sudo cp /etc/fstab /etc/fstab.orig.${epoch}
```

```
which blkid
```

```
if [ $? -eq 0 ]; then
```

```
    uuid=$(blkid | grep "${EBS_VOLUME}:" | awk '{print $2}'|sed -e 's//g')
```

```
fi
```

```
if [ -z $uuid ]; then
```

```
    id=$(lsblk -f -o NAME,UUID | grep -w "${DISK}" | awk '{print $2}'|sed -e 's//g')
```

```
    uuid="UUID=${id}"
```

```
fi
```

```
if [ ! -z $uuid ]; then
```

```
    echo "${uuid}  ${ROOST_DIR}  ext4  defaults  0  2" | sudo tee -a /etc/fstab.noroost
```

```
    sudo cp /etc/fstab.noroost /etc/fstab
```

```
    sudo umount ${ROOST_DIR}
```

```
    sudo mount -a
```

```
    if [ $? -eq 0 ]; then
```

```
        mount | grep "$EBS_VOLUME"
```

```
        if [ $? -ne 0 ]; then
```

```
            sudo chown `id -u`:`id -g` ${ROOST_DIR}
```

```
        fi
```

```
    fi
```

```
fi
```

Copy the installer and license file

1. Download Installer from Shared Google Drive <LOCATION SENT OVER EMAIL>
 - You have to download the RoostGPT installer (Roost License file will be bundled) from a shared Google Drive.
2. Transfer Installer using scp
3. Make Installer executable

```
# From your local machine, copy installer to Ubuntu instance
```

```
scp roostgpt-installer.sh ubuntu@<your-instance-ip>:~/
```

```
# Then SSH into your instance
```

```
ssh ubuntu@<your-instance-ip>
```

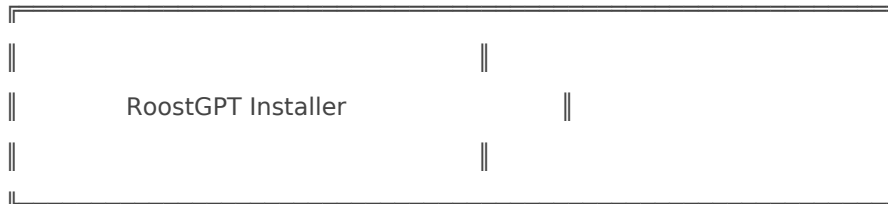
```
# Add execute permissions to installer
```

```
chmod +x roostgpt-installer.sh
```

```
# Execute Roost Installer
```

```
sudo ./roostgpt-installer.sh
```

Installation interface:



```
[→] Checking system requirements...
```

```
[✓] System requirements met
```

```
Continue with installation? [Y/n]
```

Press **Y** and **Enter** to continue.

The installer will:

- Extract files
- Install RoostGPT binary

Configuring AWS Credentials

Adding AWS Credentials

```
# SSH into your instance
ssh ubuntu@<your-instance-ip>

# Edit $HOME/.bash_profile
vim $HOME/.bash_profile

# Add export AWS keys to your $HOME/.profile and save the file
export AWS_ACCESS_KEY_ID=xxx
export AWS_SECRET_ACCESS_KEY=xxx
export AWS_DEFAULT_REGION=eu-west-1
export AWS_BEDROCK_MODEL_TYPE=cross-region # Supports [foundation, cross-region]
export AWS_BEDROCK_MODEL=global.anthropic.claude-sonnet-4-20250514-v1:0

# Source the file .profile to make the AWS creds available in the same session
source $HOME/.bash_profile
```

Using RoostGPT

Running RoostGPT

You can now access RoostGPT from anywhere on your system, using the command `roostgpt`

Quick Start Example

```
# 1. SSH into your Ubuntu instance
ssh ubuntu@<your-instance-ip>

# 2. Navigate to workspace
cd ~/workspace/functional-test

# 3. Run RoostGPT
./run.sh
```

```
# OR you can run using the full command
roostgpt test create -c functional-test.env
```

Viewing Roost Test Results

```
# Roostgpt test results can be viewed using the below command
roostgpt test result

# A sample Postman Collection with tests from a swagger spec will be generated under a folder
postman_collections/
|_ modified_postman_weather.json
|_ postman-api-test/
|_   Realtime_API.json
|_   Forecast_API.json
|_   Future_API.json
|_   History_API.json
|_   Marine_Weather_API.json
|_   Search_Autocomplete_API.json
|_   IP_Lookup_API.json
|_   Time_Zone_API.json
|_   Astronomy_API.json
```

Here's a complete workflow from login to running RoostGPT:

Configure a new test

Each project folder in your workspace contains a `<UNIQUE-TEST-NAME>.env` file for configuration. You can copy this env and modify the contents for a new test run. Or create a new test env file.

Execute roostgpt with this new env.

```
# Create a new Roost Test Configuration File
vim new_test.env

#####

# Modify the below attributes
```

```

TEST_NAME=ANY_UNIQUE_NAME # Any unique_name to represent the test configuration
TEST_TYPE=functional      # Default is unit; Supports [unit, functional, api-spec-test, integration]
AI_TYPE=bedrock_ai       # Default is openai; Supports [openai, azure_open_ai, claude_ai, dbrx, bedrock_ai,
gemini]

ROOST_USER_INPUT_TYPE=file      # Roost User input Type for test generation; Supports [text, file]
ROOST_USER_INPUT_FILE=USER_FILE_PATH # The user input file-path will be used to generate tests

# GIT env vars
GIT_TYPE=local            # Default is github; Supports [github, gitlab, azure, bitbucket, local]
LOCAL_PROJECT_PATH=PATH_TO_WORKSPACE_FOLDER # Required if GIT_TYPE is local, Path to your
workspace.

# AWS Bedrock env vars
AWS_ACCESS_KEY_ID=          # Required: If not defined here, it will be picked from environment
AWS_SECRET_ACCESS_KEY=     # Required: If not defined here, it will be picked from environment
AWS_SESSION_TOKEN=        # Optional; AWS Session token.
AWS_DEFAULT_REGION=eu-west-1 # Required if AI_TYPE=aws_bedrock.
AWS_BEDROCK_MODEL_TYPE=cross-region # Required if AI_TYPE=aws_bedrock.
AWS_BEDROCK_MODEL=global.anthropic.claude-sonnet-4-20250514-v1:0 # Required if
AI_TYPE=aws_bedrock.

# Api Spec env vars
API_SPEC_TYPE=swagger      # Optional; Supports [swagger]
API_SPEC_SOURCE=file       # Optional; Supports [file, gitpath, url]
API_SPEC_FILE_PATH="$HOME/FILE" # Optional; path of the source file.
API_SPEC_URL=              # Optional; URL of the source file, if source is url

#####

# Run the new test configuration
roostgpt test -c new_test.env

```

Troubleshooting

Installation Issues

Problem: Permission denied when running installer

```
# Solution: Make sure you're using sudo
sudo ./roostgpt-installer.sh
```

Problem: Command not found: roostgpt

```
# Solution: Verify installation completed successfully
which roostgpt

# Should output: /usr/local/bin/roostgpt

# Check executable permissions on roostgpt and grant if missing
chmod +x $(which roostgpt)

# If not found, re-run the installer
sudo ./roostgpt-installer.sh
```

Problem: Insufficient disk space

```
# Solution: Check available space
df -h $HOME /var/tmp/Roost

# Free up space or use an instance with more storage
```

Runtime Issues

Problem: Can't find workspace

```
# Solution: Workspace is in your home directory
cd ~/workspace
ls -la
```

Problem: Configuration not loading

```
# Solution: Ensure .env files exist and are readable
find ~/workspace/ -name '*.env' -ls
find ~/workspace/ -name '*.env' -exec chmod 644 {} \;
```

Verify AWS Bedrock Access

```
aws bedrock list-inference-profiles --region eu-west-1
```

Revision #24

Created 4 November 2025 14:56:10 by Harish

Updated 5 November 2025 08:39:20 by Harish