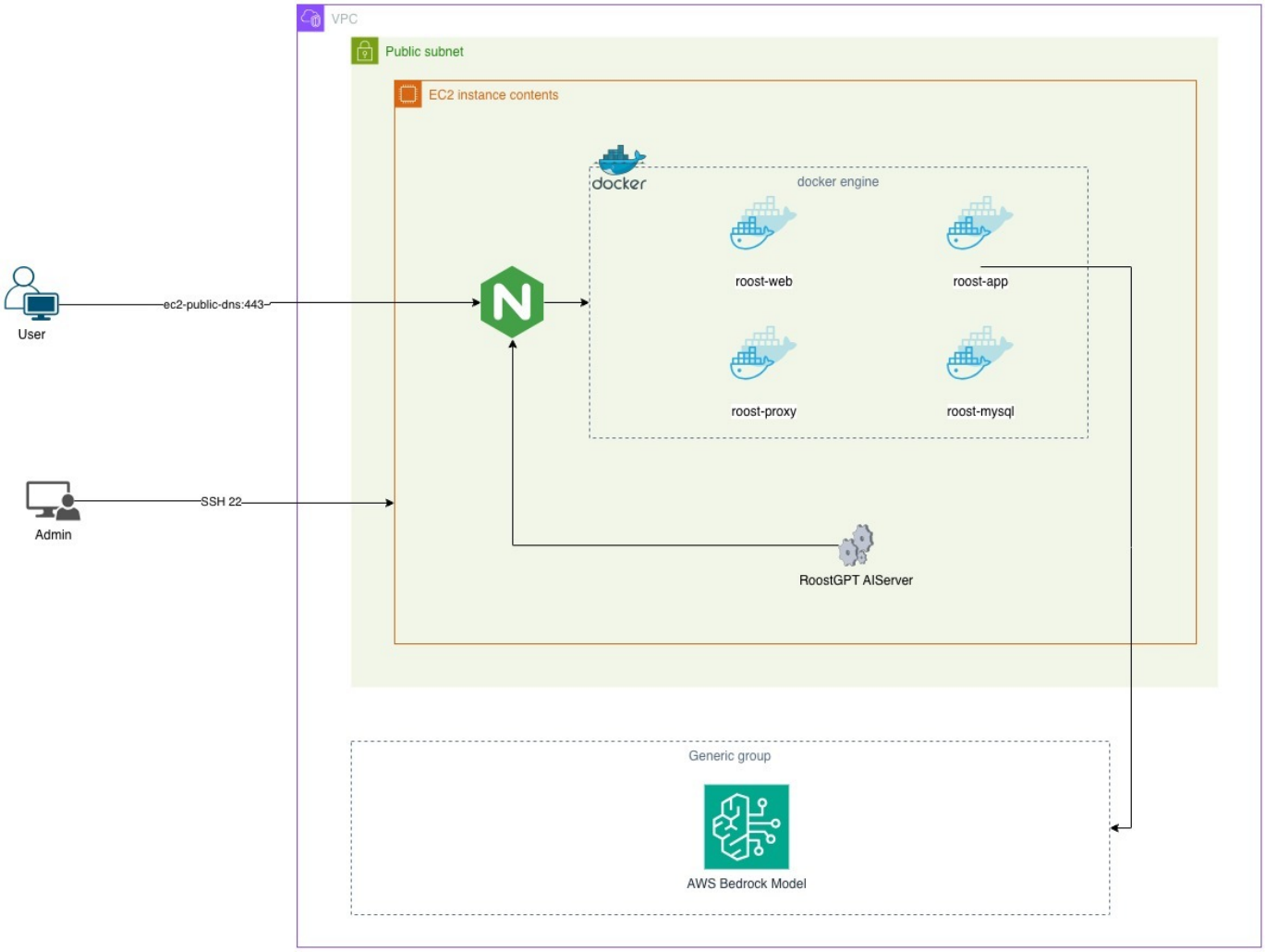
EC2 Requirements for Docker based Roost

Roost Pilot Setup (Single EC2 Instance)

Architecture of RoostGPT stack running in single instance:



Using Terraform:

To accomplish POC from single EC2 instance where RoostGPT can be setup, you can run below terraform scripts:

Link: https://github.com/roost-io/install/tree/demo/terraform/aws/bedrockdemo

Prerequisites:

- VPC ID where instance would be created
- Public Subnet ID.
- Configure AWS profile into launched server, either by setting AWS_ACCESS_KEY and AWS_SECRET_KEY_ID, in $HOME/.profile or staging $HOME/.aws/credentials
- Allowed SSH CIDR range (Could be individual IP, company network etc)

Instance Details

Whether using terraform or provisioning EC2 externally, Roost expects following configurations

- Region (eu-west-1 or any)
- Instance_size: c5a.2xlarge (16 GB Memory, 8 vCPUs) or bigger
- Instance root disk size: Minimum 100 GB
- Additional EBS disk size: Minimum 150 GB
- Image: Ubuntu 22.04 HVM base, SSD Volume Type
- Network Configuration:
  - Accepts CIDR range to allow SSH (port 22)
  - Allow SSH from 4.247.149.66/32 and 40.112.174.40/32 (Roost Support)
  - HTTPS traffic is enabled on EC2 at port 443

- Python and AWS CLI are already installed.
- If provisioned using terraform, a new ssh key-pair is created and kept under terraform-root-dir/data dir. (SSH keypair to be shared with Roost Support team)
- Default SSH user is Ubuntu which must have sudo permission
- To access RoostGPT over HTTPS (optional), we will need -
  - Domain certs
  - DNS Name.

Packages that will be installed on EC2 Linux (by roostGPT installer):

1. curl
2. jq
3. pkill
4. shasum
5. gzip
6. docker-ce
7. docker-cli

8. docker-compose
9. nginx
10. nginx-extras
11. Entry into crontab
12. Script into init.d

Configurations to run RoostGPT:

- SSH user should have sudo permissions
- AWS IAM User should have **AmazonBedrockFullAccess** or the below permission -

```
{
  Version = "2012-10-17"
  Statement = [
    {
      Effect = "Allow"
      Action = [
        "bedrock:InvokeModel",
        "bedrock:InvokeModelWithResponseStream",
        "bedrock:ListFoundationModels",
        "bedrock:GetFoundationModel"
      ]
      Resource = "*"
    },
    {
      "Effect": "Allow",
      "Action": [
          "bedrock:GetInferenceProfile",
          "bedrock:ListInferenceProfiles",
      ],
      "Resource": [
          "arn:aws:bedrock:*:*:inference-profile/*",
          "arn:aws:bedrock:*:*:application-inference-profile/*"
      ]
    }
  ]
}
```

- 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 roost license file (.ral) into EC2 instance.

- Ensure python3 is installed on the instance

Why does RoostGPT need to run with a privileged user?

- RoostGPT installs docker and nginx like services on the EC2
- RoostGPT docker containers run as root.
- RoostGPT adds crontab entry for the current user and also adds an init.d script to handle Roost processes on a m/c reboot.

Revision #4
Created 4 November 2025 05:22:16 by Harish
Updated 4 November 2025 16:36:29 by Harish