NetApp AIPod Mini para ERAG - Etapas de implantação

NetApp artificial intelligence solutions

NetApp
February 12, 2026

Índice

NetApp AIPod Mini para ERAG - Etapas de implantação . . . 1
- Pressupostos: . . . 1
- Pré-requisitos . . . 1
- Etapas de implantação do ERAG 2.0/2.0.1 . . . 2
  - 1. Baixe a versão Enterprise RAG 2.0 de GitHub . . . 2
  - 2. Instale os pré-requisitos . . . 2
  - 3. Criar arquivo de inventário . . . 2
  - 4. Configure o SSH sem senha para cada nó . . . 3
  - 5. Verifique a conectividade . . . 3
  - 6. Editar `config.yaml` arquivo . . . 4
  - 7. Implante o cluster K8s (com Trident) . . . 4
  - 8. Alterar o número de descritores abertos do iwatch . . . 5
  - 9. Instale kubectl . . . 5
  - 10. Instale o MetalLB no cluster Kubernetes . . . 5
  - 11. Configurar MetalLB . . . 5
  - 12. Atualize o arquivo config.yaml com o FQDN, o modo de acesso ao volume, o ingress e os detalhes do S3. . . . 6
  - 13. Configurar as definições de sincronização agendada . . . 8
  - 14. Implantar Enterprise RAG 2.0/2.0.1 . . . 8
  - 15. Crie uma entrada DNS . . . 8
  - 16. Acesse a interface de usuário RAG da empresa . . . 9
- Guia de solução de problemas . . . 9
  - 1. Problema: conflito de instalação do Keycloak no Helm . . . 9
  - 2. Problema: versão do Trident Operator Helm Chart não encontrada . . . 9

NetApp AIPod Mini para ERAG - Etapas de implantação

Este documento fornece um guia completo, passo a passo, para a implantação do NetApp AIPod Mini para Enterprise RAG(ERAG) 2.0. Ele abrange a instalação e configuração completamente de todos os componentes principais, incluindo a plataforma Kubernetes, NetApp Trident para orquestração de armazenamento e a pilha ERAG 2.0 usando playbooks do ansible. Além do fluxo de trabalho de implantação, o documento inclui um guia dedicado de solução de problemas que aborda problemas comuns encontrados durante a instalação, suas causas principais e resoluções recomendadas para garantir uma experiência de implantação tranquila e confiável.

Sathish Thyagarajan, Michael Oglesby, Arpita Mahajan NetApp

Pressupostos:

- O usuário responsável pela implantação possui permissões suficientes para criar namespace e instalar Helm charts.
- Os servidores Xeon executam Ubuntu 22.04.
- O mesmo nome de usuário está configurado em todos os servidores Xeon.
- O acesso administrativo ao DNS está disponível.
- ONTAP 9.16 implementado com uma SVM configurada para acesso S3.
- O bucket S3 é criado e configurado.

Pré-requisitos

Instale Git, Python 3.11 e pip para Python 3.11

No Ubuntu 22.04:

```
add-apt-repository ppa:deadsnakes/ppa
apt update
apt upgrade
apt install python3.11
python3.11 --version
apt install python3.11-pip
python3.11 -m pip --version
```

Etapas de implantação do ERAG 2.0/2.0.1

1. Baixe a versão Enterprise RAG 2.0 de GitHub

```
git clone https://github.com/opea-project/Enterprise-RAG.git
cd Enterprise-RAG/
git checkout tags/release-2.0.0
```

Para ERAG 2.0.1, use o comando abaixo

```
git checkout tags/release-2.0.1
```

2. Instale os pré-requisitos

```
cd deployment/
sudo apt-get install python3.11-venv
python3 -m venv erag-venv
source erag-venv/bin/activate
pip install --upgrade pip
pip install -r requirements.txt
ansible-galaxy collection install -r requirements.yaml --upgrade
```

3. Criar arquivo de inventário

```
cp -a inventory/sample inventory/<cluster-name>
vi inventory/<cluster-name>/inventory.ini
# Control plane nodes
kube-3 ansible_host=<control_node_ip_address>

# Worker nodes
kube-1 ansible_host=<worker_node1_ip_address>
kube-2 ansible_host=<worker_node2_ip_address>

# Define node groups
[kube_control_plane]
kube-1
kube-2
kube-3

[kube_node]
kube-1
kube-2

[etcd:children]
kube_control_plane

[k8s_cluster:children]
kube_control_plane
kube_node

# Vars
[k8s_cluster:vars]
ansible_become=true
ansible_user=<ssh_username>
ansible_connection=ssh
```

4. Configure o SSH sem senha para cada nó

```
ssh-copy-id REMOTE_USER@MACHINE_IP
```

Nota: se um nó de implantação for usado para implantar o ERAG, certifique-se de que o SSH sem senha também esteja configurado nesse nó.

5. Verifique a conectividade

```
ansible all -i inventory/<cluster-name>/inventory.ini -m ping
```

Observação: Se você não tiver o sudo sem senha configurado em seus nós, será necessário adicionar --ask -become-pass a este comando. Ao usar --ask-become-pass, é fundamental que o usuário ssh tenha a MESMA senha em cada nó.

6. Editar `config.yaml` arquivo

Prepare a implantação editando `inventory/<cluster-name>/config.yaml` para refletir as especificidades do seu ambiente.

```
vi inventory/<cluster-name>/config.yaml
```

Exemplo de trecho:

```
…
deploy_k8s: true
…
install_csi: "netapp-trident"
…
local_registry: false
…
trident_operator_version: "2510.0"    # Trident operator version (becomes
100.2506.0 in Helm chart)
trident_namespace: "trident"          # Kubernetes namespace for Trident
trident_storage_class: "netapp-trident" # StorageClass name for Trident
trident_backend_name: "ontap-nas"     # Backend configuration name
…
ontap_management_lif: "<ontap_mgmt_lif>"              # ONTAP management
LIF IP address
ontap_data_lif: "<ontap_nfs_data_lif>"                   # ONTAP data LIF
IP address
ontap_svm: "<ontap_svm>"                          # Storage Virtual Machine
(SVM) name
ontap_username: "<ontap_username>"                    # ONTAP username
with admin privileges
ontap_password: "<redacted>"                     # ONTAP password
ontap_aggregate: "<ontap_aggr>"                    # ONTAP aggregate name
for volume creation
…
kubeconfig: "<repository path>/deployment/inventory/<cluster-
name>/artifacts/admin.conf"
…
```

7. Implante o cluster K8s (com Trident)

Execute ansible-playbook playbooks/infrastructure.yaml com as tags configure e install para implantar o

cluster e Trident CSI.

```
ansible-playbook playbooks/infrastructure.yaml --tags configure,install -i
inventory/<cluster-name>/inventory.ini -e @inventory/<cluster-
name>/config.yaml
```

Nota: - Se você não tiver o sudo sem senha configurado em seus nós, será necessário adicionar `--ask-become-pass` a este comando. Ao usar `--ask-become-pass`, é fundamental que o usuário ssh tenha a MESMA senha em cada nó. - Consulte o "NetApp Trident CSI Integração para RAG Empresarial" para obter detalhes. Consulte o "Documentação de instalação do Trident" para mais detalhes.

8. Alterar o número de descritores abertos do iwatch

Consulte o "Descritores abertos do iwatch" para obter mais detalhes.

9. Instale kubectl

Consulte o "Instalar Kubectl" se ainda não estiver instalado. Recupere o arquivo kubeconfig de `<repository path>/deployment/inventory/<cluster-name>/artifacts/admin.conf`.

10. Instale o MetalLB no cluster Kubernetes

Instale o MetalLB usando helm no seu cluster Kubernetes.

```
helm repo add metallb https://metallb.github.io/metallb
helm -n metallb-system install metallb metallb/metallb --create-namespace
```

Consulte o "Instalação do MetalLB" para obter mais detalhes.

11. Configurar MetalLB

MetalLB foi configurado no modo Layer 2 e os recursos necessários IPAddressPool e L2Advertisement foram criados de acordo com as diretrizes de configuração documentadas.

```
vi metallb-ipaddrpool-l2adv.yaml
kubectl apply -f metallb-ipaddrpool-l2adv.yaml
```

Exemplo de trecho:

```
vi metallb-ipaddrpool-l2adv.yaml
---
apiVersion: metallb.io/v1beta1
kind: IPAddressPool
metadata:
  name: erag
  namespace: metallb-system
spec:
  addresses:
  - <IPAddressPool>
---
apiVersion: metallb.io/v1beta1
kind: L2Advertisement
metadata:
  name: metallb-l2adv
  namespace: metallb-system
```

Nota: - Use `metallb-system` como namespace para MetalLB IPAddressPool e L2Advertisement. - O pool de endereços IP pode incluir quaisquer IPs não utilizados na mesma sub-rede que os nós do Kubernetes. Apenas um único endereço IP é necessário para ERAG. - Consulte a "Configuração do MetalLB Layer2" para obter detalhes.

12. Atualize o arquivo config.yaml com o FQDN, o modo de acesso ao volume, o ingress e os detalhes do S3.

Modifique o arquivo config.yaml localizado em `inventory/<cluster-name>/config.yaml` para definir o FQDN de implantação, definir os modos de acesso ao volume, configurar a exposição de ingress e integrar ONTAP S3.

Edite `config.yaml` e aplique as seguintes alterações de configuração:

- FQDN: especifique o domínio totalmente qualificado usado para acessar a implantação.
- Modo de acesso ao volume: na seção gmc.pvc, configure `accessMode: ReadWriteMany` para suportar acesso simultâneo a volumes de modelo em vários pods.
- Configuração de entrada: configure o service_type de entrada como LoadBalancer para permitir o acesso externo ao aplicativo.
- Detalhes do armazenamento S3: defina storageType como s3compatible e configure os parâmetros do ONTAP S3, incluindo região, credenciais de acesso e endpoints internos e externos.
- Verificação de certificado SSL: Defina edpInternalCertVerify e edpExternalCertVerify como falso somente quando ONTAP S3 estiver configurado com certificados autoassinados. Se os certificados forem emitidos por uma CA publicamente confiável, esses parâmetros devem permanecer ativados.

Exemplo de trecho:

```
vi inventory/<cluster-name>/config.yaml
…
FQDN: "<FQDN>" # Provide the FQDN for the deployment
…
gmc:
  enabled: true
  pvc:
    accessMode: ReadWriteMany # AccessMode
    models:
      modelLlm:
        name: model-volume-llm
        storage: 100Gi
      modelEmbedding:
        name: model-volume-embedding
        storage: 20Gi
      modelReranker:
        name: model-volume-reranker
        storage: 10Gi
…
ingress:
  …
  service_type: LoadBalancer
  …
edp:
  …
  storageType: s3compatible
  …
  s3compatible:
    region: "us-east-1"
    accessKeyId: "<your_access_key>"
    secretAccessKey: "<your_secret_key>"
    internalUrl: "https://<IP-address>"
    externalUrl: "https://<IP-address>"
    bucketNameRegexFilter: ".*"
    edpExternalCertVerify: false
    edpInternalCertVerify: false
  …
```

Observação: - Por padrão, o aplicativo Intel® AI for Enterprise RAG ingere dados de todos os buckets existentes em sua SVM. Se você tiver vários buckets em sua SVM, poderá modificar o `bucketNameRegexFilter` campo para que os dados sejam ingeridos apenas de determinados buckets. - Consulte a "Implantação de RAG Intel® AI for Enterprise" documentação para obter detalhes.

13. Configurar as definições de sincronização agendada

Ao instalar o OPEA para Intel® AI for Enterprise RAG, habilite `scheduledSync` para que o aplicativo ingira automaticamente arquivos novos ou atualizados de seus buckets S3.

Quando `scheduledSync` estiver habilitado, o aplicativo verificará automaticamente seus buckets S3 de origem em busca de arquivos novos ou atualizados. Todos os arquivos novos ou atualizados encontrados como parte desse processo de sincronização são automaticamente ingeridos e adicionados à base de conhecimento do RAG. O aplicativo verifica seus buckets de origem com base em um intervalo de tempo predefinido. O intervalo de tempo padrão é 60 segundos, o que significa que o aplicativo verifica alterações a cada 60 segundos. Talvez você queira alterar esse intervalo para atender às suas necessidades específicas.

Para ativar `scheduledSync` e definir o intervalo de sincronização, defina os seguintes valores em `deployment/components/edp/values.yaml`:

```
vi components/edp/values.yaml
…
presignedUrlCredentialsSystemFallback: "true"
…
celery:
  …
  config:
    …
    scheduledSync:
      enabled: true
      syncPeriodSeconds: "60"
…
```

14. Implantar Enterprise RAG 2.0/2.0.1

Antes da instalação, valide a prontidão da infraestrutura seguindo os procedimentos descritos no "Guia de implantação do aplicativo RAG do Intel® AI for Enterprise". Esta etapa garante que a infraestrutura subjacente esteja configurada corretamente e atenda a todos os pré-requisitos necessários para uma instalação bem-sucedida do Enterprise RAG Application.

Execute a instalação usando:

```
ansible-playbook -u $USER playbooks/application.yaml --tags
configure,install -e @inventory/<cluster-name>/config.yaml
```

Observação: se você não tiver o sudo sem senha configurado no seu nó de implantação (o laptop ou host de acesso remoto onde você está executando o comando ansible-playbook), será necessário adicionar `--ask-become-pass` a este comando. Ao usar `--ask-become-pass`, é fundamental que o usuário ssh tenha a MESMA senha em cada nó.

15. Crie uma entrada DNS

Crie uma entrada DNS para o painel de controle web do Enterprise RAG em seu servidor DNS. Para

prosseguir, recupere o endereço IP externo atribuído à entrada do Enterprise RAG LoadBalancer:

```
kubectl -n ingress-nginx get svc ingress-nginx-controller
```

Crie uma entrada DNS apontando para este endereço IP para o FQDN que você usou na Etapa 12.

Nota: - O FQDN usado para a entrada DNS DEVE corresponder ao FQDN do arquivo de configuração.

16. Acesse a interface de usuário RAG da empresa

Acesse a interface de usuário do Enterprise RAG navegando até o FQDN correspondente no seu navegador. Observação: você pode recuperar as credenciais padrão da interface de usuário em cat ansible-logs/default_credentials.txt

Guia de solução de problemas

1. Problema: conflito de instalação do Keycloak no Helm

Cenário: Durante a implantação do ERAG, a instalação do Keycloak pode falhar com o seguinte erro:

```
FAILED - RETRYING: [localhost]: Install Keycloak Helm chart (5 retries
left).
Failure when executing Helm command. Exited 1.
    stdout:
    stderr: Error: UPGRADE FAILED: another operation
(install/upgrade/rollback) is in progress
```

Ação: Se a falha persistir após novas tentativas – desinstale a implantação do ERAG, exclua o namespace de autenticação existente usando os comandos abaixo e execute a implantação novamente.

```
ansible-playbook playbooks/application.yaml --tags uninstall -e
@inventory/<cluster-name>/config.yaml

helm -n auth uninstall keycloak
kubectl -n auth get pvc # confirm all PVCs are gone; if any are left,
delete them
kubectl delete ns auth
```

Observação: um estado de versão desatualizado do Helm pode bloquear operações subsequentes de instalação ou atualização.

2. Problema: versão do Trident Operator Helm Chart não encontrada

Cenário: Durante a implantação do ERAG, a instalação do Trident pode falhar devido a uma incompatibilidade de versão do Helm chart. O seguinte erro pode ser observado:

```
TASK [netapp_trident_csi_setup : Install Trident operator via Helm]
fatal: [localhost]: FAILED! => changed=false
  command: /usr/local/bin/helm --version=100.2510.0 show chart 'netapp-
trident/trident-operator'
  msg: |-
    Failure when executing Helm command. Exited 1.
    stdout:
    stderr: Error: chart "trident-operator" matching 100.2510.0 not found
in netapp-trident index.
            (try 'helm repo update'): no chart version found for trident-
operator-100.2510.0
```

Ação: se esse erro ocorrer, atualize o índice do repositório Helm e execute novamente o playbook de implantação.

```
helm repo update
ansible-playbook playbooks/application.yaml -e @inventory/<cluster-
name>/config.yaml
```

Nota: este é um problema conhecido na versão 2.0 do ERAG. Uma correção foi enviada e será incluída em uma versão futura.

Informações sobre direitos autorais

Informações sobre marcas comerciais