**Apache Airflow 3 On-Prem Deployment Configuration (RHEL 8/9)**

This document provides step-by-step configuration details for each machine in your RHEL 8/9 on-premise Airflow 3 production environment.

**1. Common Pre-requisites (All Nodes)**

1. **Create airflow user**:
2. sudo useradd -m -s /bin/bash airflow
3. **Enable EPEL and Required Repos**:
4. sudo dnf install -y epel-release
5. sudo dnf config-manager --set-enabled powertools || true
6. **Install Basic Utilities**:
7. sudo dnf install -y git vim wget curl unzip which policycoreutils-python-utils
8. **SELinux Configuration**:
   * Ensure SELinux in enforcing mode.
   * For NFS mounts and SSH keys, apply contexts as detailed per role.

**2. NFS Server (Shared DAGs & SSH Config)**

* **Packages**:
* sudo dnf install -y nfs-utils
* **Export Directory**:
* sudo mkdir -p /srv/airflow-nfs/{dags,plugins,ssh}
* sudo chown -R airflow:airflow /srv/airflow-nfs
* **/etc/exports**:
* /srv/airflow-nfs \*(rw,sync,no\_root\_squash)
* **Enable & Start**:
* sudo systemctl enable --now nfs-server
* sudo exportfs -rav

**3. Metadata DB Server (PostgreSQL + Patroni)**

* **Packages**:
* sudo dnf install -y postgresql-server postgresql-contrib
* **Initialize DB** (RHEL 9):
* sudo postgresql-setup --initdb
* **Configure postgresql.conf**:
* listen\_addresses = '\*'
* max\_connections = 200
* wal\_level = replica
* archive\_mode = on
* archive\_command = 'cd .'
* **Configure pg\_hba.conf**:
* host all all 10.0.0.0/24 md5
* **Create Airflow DB/User**:
* sudo -iu postgres psql <<EOF
* CREATE USER airflow WITH PASSWORD 'StrongPass123';
* CREATE DATABASE airflow OWNER airflow;
* EOF
* **Enable & Start**:
* sudo systemctl enable --now postgresql

*(Optionally install Patroni for HA)*

**4. Message Broker Server (RabbitMQ)**

* **Enable RabbitMQ Repo**:
* sudo dnf install -y https://packagecloud.io/rabbitmq/rabbitmq-server/packages/el/9/noarch/rabbitmq-server-3.11.x.el9.noarch.rpm
* **Install & Start**:
* sudo dnf install -y rabbitmq-server
* sudo systemctl enable --now rabbitmq-server
* **Add Airflow User/VHost**:
* sudo rabbitmqctl add\_vhost /airflow
* sudo rabbitmqctl add\_user airflow StrongPass123
* sudo rabbitmqctl set\_permissions -p /airflow airflow ".\*" ".\*" ".\*"
* **TLS (Optional)**:
  + Place certs in /etc/rabbitmq/ and configure in /etc/rabbitmq/rabbitmq.conf.

**5. API Server Nodes (2x)**

* **Mount NFS**:
* sudo mkdir -p /mnt/airflow
* sudo mount -t nfs nfs-server:/srv/airflow-nfs /mnt/airflow
* echo "nfs-server:/srv/airflow-nfs /mnt/airflow nfs defaults 0 0" | sudo tee -a /etc/fstab
* **Install Python & Airflow**:
* sudo dnf install -y python3 python3-venv gcc openssl-devel libffi-devel
* sudo -iu airflow bash -c '
* python3 -m venv ~/venv
* source ~/venv/bin/activate
* pip install --upgrade pip
* pip install "apache-airflow[celery,postgres,crypto]==2.11.0"
* '
* **airflow.cfg** (partial):
* [api]
* auth\_backend = airflow.api.auth.backend.default
* [core]
* executor = CeleryExecutor
* sql\_alchemy\_conn = postgresql+psycopg2://airflow:StrongPass123@db-server:5432/airflow
* [celery]
* broker\_url = amqp://airflow:StrongPass123@broker-server:5672/airflow
* result\_backend = db+postgresql://airflow:StrongPass123@db-server:5432/airflow
* **Systemd Service** (/etc/systemd/system/airflow-api.service):
* [Unit]
* Description=Airflow API Server
* After=network.target
* [Service]
* User=airflow
* Environment="PATH=/home/airflow/venv/bin"
* ExecStart=/home/airflow/venv/bin/airflow api serve
* Restart=on-failure
* [Install]
* WantedBy=multi-user.target
* **Enable & Start**:
* sudo systemctl daemon-reload
* sudo systemctl enable --now airflow-api

**6. Webserver Nodes (2x)**

*(Identical to API nodes, plus webserver service)*

* **Mount NFS & Python Setup**: Same as API servers.
* **airflow.cfg** (add webserver section):
* [webserver]
* web\_server\_host = 0.0.0.0
* web\_server\_port = 8080
* rbac = True
* auth\_backend = airflow.www.auth.backend.default
* **Systemd Service** (airflow-webserver.service):
* [Unit]
* Description=Airflow Webserver
* After=network.target
* [Service]
* User=airflow
* Environment="PATH=/home/airflow/venv/bin"
* ExecStart=/home/airflow/venv/bin/airflow webserver
* Restart=on-failure
* [Install]
* WantedBy=multi-user.target
* **Start**:
* sudo systemctl enable --now airflow-webserver

**7. Scheduler Nodes (2x)**

* **Mount NFS & Python Setup**: Same as API servers.
* **airflow.cfg** Scheduler Settings:
* [scheduler]
* job\_heartbeat\_sec = 5
* scheduler\_heartbeat\_sec = 5
* max\_threads = 4
* **Systemd Service** (airflow-scheduler.service):
* [Unit]
* Description=Airflow Scheduler
* After=network.target
* [Service]
* User=airflow
* Environment="PATH=/home/airflow/venv/bin"
* ExecStart=/home/airflow/venv/bin/airflow scheduler
* Restart=on-failure
* [Install]
* WantedBy=multi-user.target
* **Start**:
* sudo systemctl enable --now airflow-scheduler

**8. TEI & Worker Nodes (can co-locate) (2x+)**

* **Mount NFS & Python Setup**: Same NFS mount and venv.
* **SSH Key & Known Hosts**:
* sudo mkdir -p /home/airflow/.ssh
* sudo cp /mnt/airflow/ssh/airflow\_deploy\_key /home/airflow/.ssh/id\_ed25519
* sudo cp /mnt/airflow/ssh/known\_hosts /home/airflow/.ssh/known\_hosts
* sudo chown -R airflow:airflow /home/airflow/.ssh
* sudo chmod 600 /home/airflow/.ssh/id\_ed25519
* **Airflow Connection** (airflow connections add):
* airflow connections add ssh\_shell\_servers \
* --conn-type SSH \
* --conn-login airflow-runner \
* --conn-port 22 \
* --conn-extra '{"key\_file": "/home/airflow/.ssh/id\_ed25519"}'
* **Systemd Services**: Create both TEI and worker services:
  + airflow-tei.service:
  + [Unit]
  + Description=Airflow Task Execution Interface
  + After=network.target
  + [Service]
  + User=airflow
  + Environment="PATH=/home/airflow/venv/bin"
  + ExecStart=/home/airflow/venv/bin/airflow tasks execute-server
  + Restart=on-failure
  + [Install]
  + WantedBy=multi-user.target
  + airflow-worker.service:
  + [Unit]
  + Description=Airflow Celery Worker
  + After=network.target
  + [Service]
  + User=airflow
  + Environment="PATH=/home/airflow/venv/bin"
  + ExecStart=/home/airflow/venv/bin/airflow celery worker
  + Restart=on-failure
  + [Install]
  + WantedBy=multi-user.target
* **Enable & Start**:
* sudo systemctl daemon-reload
* sudo systemctl enable --now airflow-tei airflow-worker

**9. Monitoring & Logging Nodes**

* **Prometheus & Node Exporter**:
* sudo dnf install -y prometheus prometheus-node-exporter
* sudo systemctl enable --now prometheus prometheus-node-exporter
* **Grafana**:  
  Install via RPM from grafana.com and configure dashboards to scrape Prometheus.
* **ELK Stack**:
* sudo dnf install -y elasticsearch kibana logstash
* sudo systemctl enable --now elasticsearch kibana logstash
* **Configure Airflow** (airflow.cfg):
* [logging]
* remote\_logging = True
* remote\_log\_conn\_id = elastic
* remote\_base\_log\_folder = "elasticsearch://localhost:9200/logs"

**10. Backup & Disaster Recovery**

* **PostgreSQL Dump Cron** (/etc/cron.daily/pg\_backup):
* #!/bin/bash
* PGPASSFILE=/root/.pgpass
* pg\_dump -h db-server -U airflow airflow | gzip > /backup/airflow\_$(date +%F).sql.gz
* # /root/.pgpass
* db-server:5432:airflow:airflow:StrongPass123
* **RabbitMQ Snapshot**:
* rabbitmqctl export\_definitions /backup/rabbit\_defs\_$(date +%F).json
* **NFS Volume Snapshot**: Use LVM or SAN snapshot tools nightly.

*End of configuration guide.*