The Why and How of Running a Self-Managed Airflow on Kubernetes

Parnab Basak (he/him)
Senior Solution Architect and Airflow SME
Amazon Web Services
Agenda
for the next 20 mins

- Available deployment options
- Why should you need to run self-managed Airflow
- How to deploy Airflow on Kubernetes using automation
- Developer & Operator experience
- Owned Responsibilities and tradeoffs
- Questions?
Available Deployment Options

On-premises

Sources on Server
Containers
Containers with Kubernetes
Virtual Machines

Cloud

AWS
Amazon EC2
Amazon EKS
Amazon MWAA

Microsoft Azure
Azure VMs
Azure Kubernetes Service
Azure Data Factory

Google Cloud
Google Compute Engine
Google Kubernetes Engine
Google Cloud Composer

Personal Devices

docker + kubernetes

ASTRONOMER
Quube

...Others
Self-Managed vs Platform as a Service

The diagram compares the concepts of self-managed and platform-as-a-service (PaaS) environments. It highlights the following services and tools:

**Sources**
- Amazon MWAA
- Google Cloud Composer
- Astronomer Astro
- Azure Data Factory
- Quubele

**Containers**
- Container with Orchestration

**Our Focus**
- Self-Managed
- Platform as a Service
Advantages with Airflow as a Service

- Setup
- Security
- Scaling
- Maintenance
- High Availability
- Upgrades
Why Self-Managed
# Reasons for Self-Managed Airflow

## Flexibility and Choices
- Available Versions
- Diff Executor/Database
- Custom Code

## Feature Gaps
- Parity with Open-Source
- Abstractions

## Avoid Lock-ins
- On-Prem / Non-Cloud / Non-Docker
- Contracts/Agreement

## Infrastructure Costs
- Re-use Existing Infra
- Savings / Discounts
- Pause / Freeze

## Security
- Data Sovereignty
- Internal Governance

## Compliance
- FedRAMP/CJIS
- HIPAA / PCI
- SOC
How to install Self-Managed
# How to Self Install Apache Airflow

<table>
<thead>
<tr>
<th>How</th>
<th>Released Sources</th>
<th>PyPi</th>
<th>Docker Images</th>
<th>Kubernetes</th>
</tr>
</thead>
<tbody>
<tr>
<td>How</td>
<td>Download Code &gt; Build Code &gt; Install</td>
<td>Build &amp; Deploy everything</td>
<td>Download Images &gt; Customize &gt; Deploy</td>
<td>Download &amp; install via Helm charts</td>
</tr>
<tr>
<td>When</td>
<td>Code build and all component/dependency configuration</td>
<td>All component/dependency configuration</td>
<td>Familiar with Docker/Containers. Run Airflow in isolation</td>
<td>Familiar with Docker/Containers &amp; Kubernetes. Run Airflow in isolation</td>
</tr>
<tr>
<td>What to Handle</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Effort</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★</td>
<td>★</td>
</tr>
</tbody>
</table>
Why Kubernetes

- Open Source
- Rich Eco System
- Faster Time to Market

https://www.bestdevops.com/easy-way-to-know-kubernetes/
Installing **Apache Airflow on Amazon EKS - Helm**

1. eksctl
   - deployment spec file
   - VPC/Subnet
   - NodeGroup
   - Namespace
   - Create

2. Amazon EKS
   - Cluster created
   - Update
   - Amazon EKS with cluster autoscaler
   - Deploy
   - RDS PostgresSQL
   - Pull

3. Amazon EFS
   - Push
   - Amazon ECR

4. Apache Airflow on EKS
   - with HorizontalPodAutoscaler
   - Access Airflow UI

---

**AWS Cloud**
Developer & Operator Experience

AWS Cloud

CI/CD Account

Uses Image

Uses Image

Amazon ECR

Pull dev branch

Airflow DAGs

Pull main branch

Airflow Image Spec

Git - Sync sidecar

Git - Sync sidecar

Dev Account

Dev Airflow Cluster

PROD Account

Production Airflow Cluster

Watches

GiTBH

Git-Sync sidecar

ARGOCD

Airflow ArgoCD
**Operator Observability**

- **Kubernetes Cluster**
  - Airflow Instance
    - Webserver
    - Scheduler
    - Worker
  - Push Metrics to StatsD
  - Scrape Metrics from StatsD
- Prometheus
  - Display Metrics
- Grafana
- Admin

Options:
- argo
- HELM
Self-Managed Responsibility

Knowledge
- Apache Airflow
- Kubernetes
- Networking

Responsibility
- Setup, Upgrade, Maintenance
- AuthZ and AuthN
- High Availability
- Autoscaling, CI/CD
- Logging
- Observability

Gain
- Airflow with Custom Capabilities

...Just to name a few
Key Takeaways

**Key Requirements**
- List and Prioritize
- Must Haves
- Nice to Haves

**Evaluate MSP**
- Value Propositions
- Features Supported
- Roadmap

**Decide Hosting**
- Where
- How
- Why
- Owned Responsibilities

**Automation**
- What / How (Re-use)
- Scale
- Maintainability
Session Survey

https://www.pulse.aws/survey/STI90ZJG

- SIX Questions = Less than a Minute
- Completely Anonymous

- 1 star – least favorable
- 5 star – most favorable
Questions?

https://www.linkedin.com/in/parnab-basak/