Overview

This document describes the components of Cerebro Data Access Service. It provides an overview of what each component does, their requirements for deployment and how typical user deployments could look like and their tradeoffs. This document also describes how to use and configure infrastructure details regarding networking, compute and storage.

Cerebro Data Access Service (CDAS) provides a platform service to enable unified and secure data access for existing storage systems. Enterprises that need to store more data, enable analytics with more diverse access paths and audit and control data access at fine granularity often find it difficult to satisfy all of these requirements with the existing technologies. Newer storage systems (e.g. AWS S3) that provide the right capabilities in terms of cost and scalability do not support the required rich access semantics (e.g. sub-file access controls). Enforcing the access policies in the compute tools is similarly challenging as it greatly limits the ways that end users can work with the data. CDAS solves this problem by providing a service that can serve as a single access point which abstracts away the storage system behind the scene and provides an easy to use set of APIs that analytics layers can use to enable user workflows.

Service components

CDAS consists of three components that operators will managed:

1. Deployment Manager: This is the deployment administration service which is responsible for managing all of the other CDAS components in your enterprise. It provides capabilities to spin up and down other service and monitor their status.
2. Catalog Service: This service maintains and provides all metadata for data managed by CDAS. The metadata includes data schemas and access policies.
3. Data Service: This service is responsible for delivering data to the end tools, enforcing any access policies. It reads data from the underlying storage systems and does all of the required data transformations. This service requires a catalog service to be running.

All services provide a REST API. The Deployment Manager and CatalogService also provide CLI and web GUI interfaces.

All components run as Docker containers in the user environment. The catalog service and data access service are each distributed. Each service is expected to run in a single data center but different instances of the service can run in different environments. The Deployment Manager and Catalog Service are currently required to run in AWS, with a typical deployment having them run on their own instances. The Data Access Service can be run in any infrastructure that can run containers.[1]

Deployment Architecture

A typical CDAS deployment consists of the following:

- 1x DeploymentManager instance
- 1x CatalogService instance
- 1x long running DataService instance
- 1x (or more) transient DataService instances that are created on demand

For the purpose of this document, we'll describe the deployment architecture in AWS. The Cerebro platform uses and depends on the following AWS services:

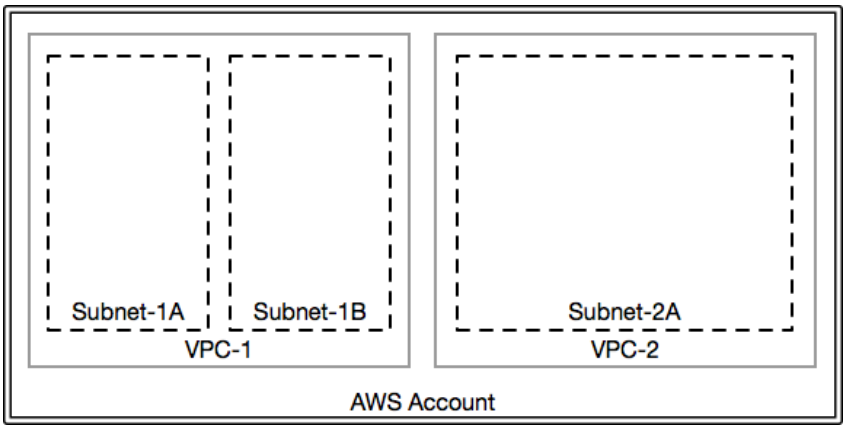
- EC2
    - Images can be provided by Cerebro or the enterprise. It must be able to run docker containers.
- RDS (to back the CatalogService and DeploymentManager)
- S3 (to store logs and config backups)

[1] In the initial beta release, all services are expected to be in AWS running in the same region. To run containers, the EC2 container service is sufficient.

In RDS and S3, Cerebro will store data that is not expected to be writable by systems outside of the Cerebro stack. The database is expected to only be accessible by the CatalogService and DeploymentManager.

Network Topology

In a typical AWS account, you would have one more VPCs and each VPC would have one or more subnets with different levels of network connectivity and permissions. In any organization, you could have more than one AWS account, with each having multiple VPCs and subnets.



There are 3 models for deploying Cerebro:

1. The complete Cerebro stack deployed inside a single subnet in a single VPC.
2. The DeploymentManager and CatalogService deployed in their own subnets and each instance of the DataService deployed in independent subnets -- all within the same VPC. This model gives finer grained control over the networking in the different subnets and who gets to access what pieces.
3. The DeploymentManager and CatalogService deployed in their own subnets in one VPC. Each instances of the DataService deployed in different VPCs or even different accounts. This model is suitable for scenarios where different accounts or VPCs are owned by different teams.

As long as the following requirements are met, the details of the network do not matter:

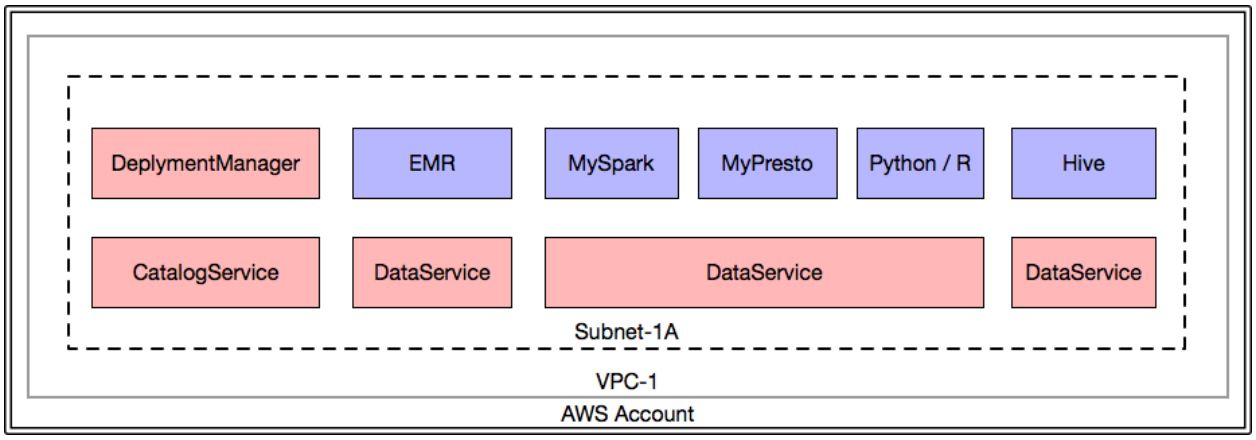
- DeploymentManager needs connectivity with CatalogService and DataService
- DataService needs connectivity to the CatalogService
- DataService instances do not need connectivity with each other.

Model #1

A complete CDAS deployment could be contained within the confines of a single VPC, single subnet as shown in the following diagram. CDAS instances don’t need a public IP address and can run within a private subnet since no external network connections are required but the

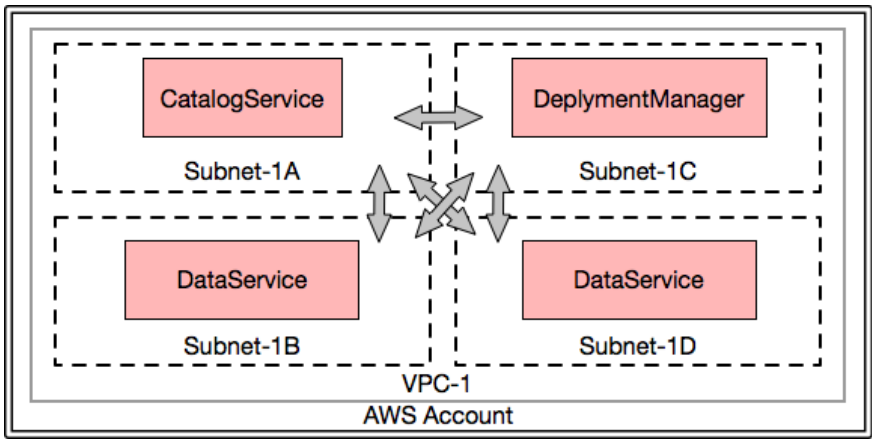
internal components will need to communicate with each other. Communication can be restricted to a few predefined ports if necessary. Each CDAS component can deploy within a single subnet and have their own security groups depending on how you'd like to restrict access to the different components. They can also deploy within a single security group. Both are valid deployment options.

The Cerebro DataService will deploy in its own security groups with restricted access to admins and opening ports required for query layers to integrate with. The query layers will deploy in their own security groups, if they are running in the same AWS account. They can also run in different AWS accounts (for individual business units / groups) and access the Cerebro components as long as connectivity is set up between that VPC and the VPC where Cerebro components are running.

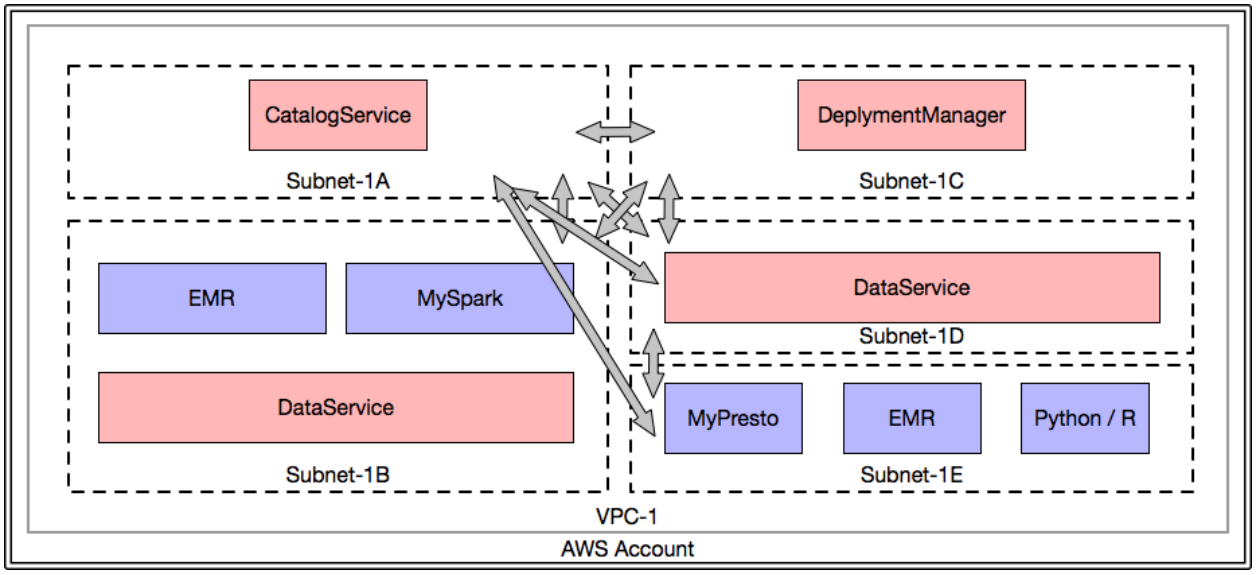


Model #2

This model entails having multiple subnets within a single VPC. There's dedicated subnets for each component of the Cerebro stack. There can be one or more instances of the DataService running, each in it's own subnet. The different DataService instances don't need access to each other. This topology is shown in the figure below.



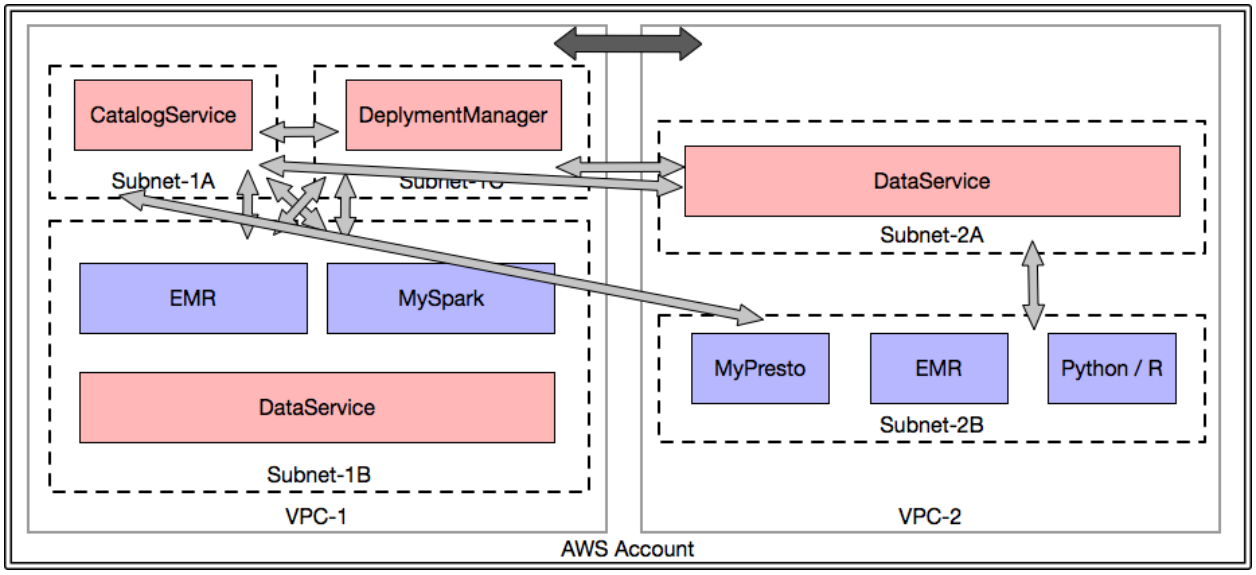
The services that access the DataService (eg: EMR, Databricks) could be running in the same subnet or in different subnets, as long as they have full connectivity to the DataService instance and the CatalogService. This is shown in the following figure:



Model #3

In this model, you can have multiple VPCs, potentially for different applications or business units and deploy instances of the DataService in different VPCs, with the CatalogService and DeploymentManager running in a single centralized VPC. The intent is to give people self contained environments that they can work with, while doing the administration from a central environment. The different VPCs need to have connectivity to the central VPC where the DeploymentManager and CatalogService are running. This can be accomplished with VPC Peering[2].

[2] VPC Peering: http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-peering.html



Security Groups and Ports

The CatalogService, DeploymentManager and DataService should be able to communicate with each other. The ports for these are set in configurations at the time of installation.

Sizing recommendations

DeploymentManager

DeploymentManager is expected to run on one machine and can simply be restarted if necessary. It is not required to access data, only to provision additional instances. The state of the deployments and environments will be stored in RDS. The instance itself doesn't hold any state. We recommend running the DeploymentManager on a m4.2xlarge instance.

CatalogService

CatalogService can be run at most scale for high availability. A typical size of 3 or 5 nodes is sufficient. m4.2xlarge instances are good enough for this as well.

DataService

DataService should be run at the same scale as the compute instances. This should use similar instances to the compute nodes. The DataService needs nominal memory and CPU. Instances like m4.xlarge, m3.xlarge or c3.xlarge are appropriate choices for the DataService deployments.

User and Identity Management

Service Credentials

The Cerebro components need 2 static IAM users in the AWS account that will host the various components. One IAM user (referred to as IAM_Mgmt here) is for the DeploymentManager and the CatalogService and the second IAM user (referred to as IAM_DAS here) is for the DataService instances.

The IAM_Mgmt user should be able to do the following:

- Provision EC2 instances
- Provision EC2 containers
- Provision RDS instances
- Full read/write access on the logs S3 bucket

The IAM_DAS user should be able to do the following:

- Full read/write access on the data lake S3 bucket
- Full read/write access on the logs S3 bucket

User Credentials

Users for the Cerebro install come from AD/LDAP, which could be setup as local unix users and groups on the servers where Cerebro is installed. Users authenticate with the components initially with kerberos which will return a token that is used for all subsequent requests.