CDP Public Cloud

AWS Reference Network Architecture

Date published: 2019-08-22
Date modified:

https://docs.cloudera.com/

Legal Notice

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

CDP Public Cloud reference network architecture for AWS

This topic includes a conceptual overview of the CDP Public Cloud network architecture for AWS, its use cases, and personas who should be using it

Overview

CDP Public Cloud allows customers to set up cloud Data Lakes and compute workloads in their cloud accounts on AWS, Azure, and Google Cloud. It maps a cloud account to a concept called the environment into which all compute workload clusters (Data Hubs) and data services (such as Cloudera Data Engineering (CDE), Cloudera Data Warehouse (CDW), Cloudera Machine Learning (CML), Cloudera Operational Database (COD), Cloudera DataFlow (CDF)) are launched. For these Data Lakes, compute workload clusters, and data services to function correctly, several elements of the cloud architecture need to be configured appropriately: access permissions, networking setup, cloud storage and so on. Broadly, these elements can be configured in one of two ways:

- CDP can set up these elements for the customer

  Usually, this model helps to set up a working environment quickly and try out CDP. However, many enterprise customers prefer or even mandate specific configurations of a cloud environment for Infosec or compliance reasons. Setting up elements such as networking and cloud storage requires prior approvals and they would generally not prefer, or even actively prevent, a third party vendor like Cloudera to set up these elements automatically.
- CDP can work with pre-created elements provided by the customer

  In this model, the flow for creating the cloud Data Lakes accepts pre-created configurations of the cloud environment and launches workloads within those boundaries. This model is clearly more aligned with enterprise requirements. However, it brings with it the risk that the configuration might not necessarily play well with CDP requirements. As a result, customers might face issues launching CDP workloads and the turnaround time to get to a working environment might be much longer and involve many tedious interactions between Cloudera and the customer cloud teams.

From our experience in working with several enterprise customers, the most complicated element of the cloud environment setup is the cloud network configuration. The purpose of this document is to clearly articulate the networking requirements needed for setting up a functional CDP Public Cloud environment into which the Data Lakes and compute workloads of different types can be launched. It attempts to establish the different points of access to these workloads and establishes how the given architecture helps to accomplish this access.

Along with this document, you can use the cloudera-deploy tool to automatically set up a model of this reference architecture, which can then be reviewed for security and compliance purposes.

Use cases

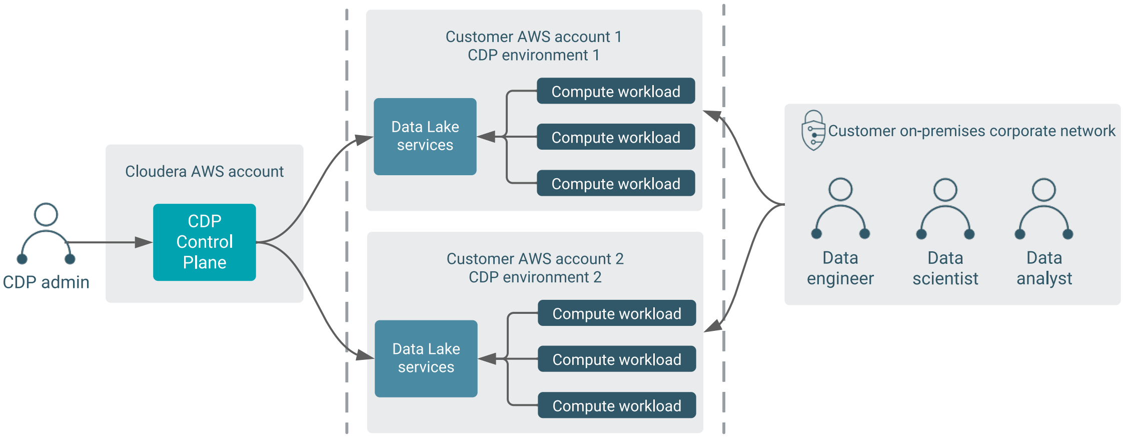
CDP Public Cloud allows customers to process data in the cloud storage under a secure and governed Data Lake using different types of compute workloads that are provisioned via Data Hub or data services. Typically the lifecycle of these workloads is as follows:

- A CDP environment is set up by a CDP admin using their cloud account. This sets up a cloud Data Lake cluster and FreeIPA cluster with security and governance services and an identity provider for this environment. The CDP admin may need to work with a cloud administrator to create all the cloud provider resources (including networking resources) that are required by CDP.
- Then one or more compute workload clusters can be launched, linked to the Data Lake. Each of these workload clusters typically serves a specific purpose such as data ingestion, analytics, machine learning and so on.
- These compute workload clusters are accessed by data consumers like data engineers, analysts or scientists. This is the core purpose of using CDP on the public cloud.
- These compute workload clusters can be long-running or ephemeral, depending on the customer needs.

There are two types of users for CDP who interact with the product for different purposes:

- CDP admins - These persons are usually concerned with the launch and maintenance of the cloud environment, and the Data Lake, Data Hubs, FreeIPA, and CDP data services running inside the environment. They use a Management Console running in the Cloudera AWS account to perform these operations of managing the environment.
- Data consumers - These are the data scientists, data analysts, and data engineers who use the Data Hubs and data services to process data. They mostly interact directly with the compute workloads (Data Hubs and data services) running in their cloud account. They could access these either from their corporate networks (typically through a VPN) or other cloud networks their corporate owns.

These two types of users and their interaction with CDP are represented in the following diagram:



Taxonomy of network architectures

This topic provides a high-level overview of each type of network architecture that CDP supports.

At a high level, there are several types of network architectures CDP supports. As can be expected, each type brings a unique trade-off among various aspects, such as ease of setup, security provided, workloads supported, and so on. This section only provides a high level overview of each type. The characteristics of each type are explained under appropriate sections in the rest of the document. The users must review the advantages and disadvantages of each of these taxonomies in detail before making a choice suitable to their needs.

Name	Description	Trade-offs
Publicly accessible networks	Deploys customer workloads to hosts with public IP addresses. Security groups must be used to restrict access only to corporate networks as needed.	Easy to set up for POCs. Low security levels.
Semi-private networks	Deploys customer workloads to private subnets, but exposes services to which data consumers need access over a load balancer with a public IP address. Security groups or allow-lists (of IP addresses or ranges) on load balancers must be used to restrict access to these public services only to corporate networks as needed.	This option is fairly easy to set up too, but it may not solve all the use cases of access (in semi private networks). The surface of exposure is reduced, and it is reasonably secure.
Fully private networks	Deploys customer workloads to private subnets and even services to which data consumers need access are only on private IPs. Requires connectivity to corporate networks to be provided using solutions like VPN gateways, and so on.	Complex to set up depending on prior experience of establishing such connectivity, primarily due to the way the customer has to solve the corporate network peering problem. But it is very secure.

Name	Description	Trade-offs
Fully private outbound restricted networks	This is the same as fully private networks; Except, in addition, Cloudera provides a mechanism for users to configure an outbound proxy or firewall to monitor or restrict the communication outside their networks.	Most complex to set up, mainly considering the varied needs that data consumers would have to connect outside the VPC on an evolving basis. It is also the most secure for an enterprise.

Management Console to customer cloud network

This topic explains the possible ways in which CDP Control Plane can communicate with the compute infrastructure in the customer network, in the context of the Management Console.

As described previously, the CDP admin would typically use the CDP Management Console that runs in the CDP Control Plane to launch CDP environments with Data Lakes, FreeIPA, Data Hubs, and data services into their cloud accounts. In order to accomplish this, the CDP Control Plane and the compute infrastructure in the customer network (such as EC2 instances, EKS clusters) should be able to communicate with each other. Depending on the chosen network architecture, this communication can occur in the ways described below.

Publicly accessible networks

In this model of publicly accessible networks, the compute infrastructure must be reachable over the public internet from the Management Console. While this is fairly easy to set up, it is usually not preferred by enterprise customers, as it implies that the EC2 nodes or EKS nodes are assigned public IP addresses. While the access control rules for these nodes can still be restricted to the IP addresses of the Management Console components, it is still considered insecure for each of the network architectures described earlier.

Semi-private networks

Publicly accessible networks are easy to set up for connectivity, both from the CDP Control Plane and the customer on-prem network, but have a large surface area of exposure as all compute infrastructure has public IP addresses. In contrast, fully private networks need special configuration to enable connectivity from the customer on-prem network, due to having no surface area of exposure to any of the compute infrastructure. While very secure, it is more complex to establish.

There is a third configuration supported by CDP, semi-private networks, that provides some trade-offs between these two options. In this configuration, the user deploys the worker nodes of the compute infrastructure on fully private networks as described above. However, the user chooses to expose UIs or APIs of the services fronting these worker nodes over a public network load balancer. By using this capability, the data consumers can access the UIs or APIs of the compute infrastructure through these load balancers. It is also possible to restrict the IP ranges from which such access is allowed using security groups.

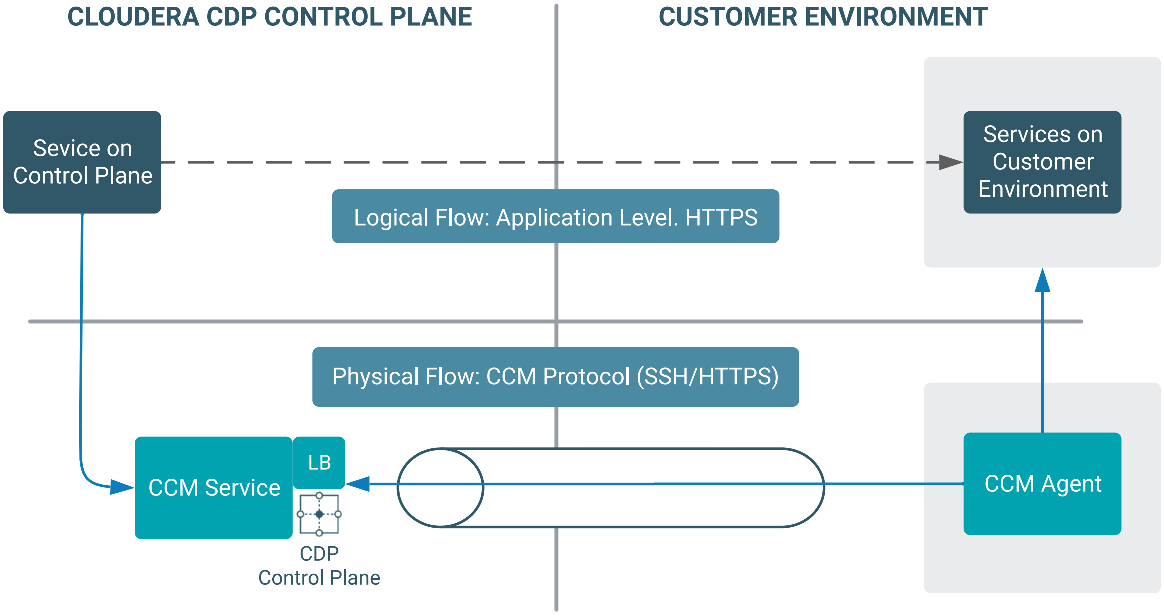
While this option provides a trade-off between ease of setup and exposure levels, it may not satisfy all use cases related to communication between various endpoints. For example, some compute workloads involving Kafka or NiFi would not benefit from having a simple publicly exposed NLB. It is recommended that customers evaluate their use cases against the trade-off and choose an appropriately convenient and secure model of setup.

Fully private networks

In this model of fully private networks, the compute infrastructure is not assigned any public IP addresses. In this case, communication between the CDP Control Plane and compute infrastructure is established using a "tunnel" that originates from the customer network to the CDP Control Plane. All communication from the CDP Control Plane to the compute nodes is then passed through this tunnel. From experience, Cloudera has determined that this is the preferred model of communication for customers.

To elaborate on the tunneling approach, Cloudera uses a solution called Cluster Connectivity Manager (CCM). At a high level, the solution uses two components, an agent (CCM agent) that runs on a VM provisioned in the customer network and a service (CCM service) that runs on the CDP Control Plane. The CCM agent, at start-up time, establishes a connection with the CCM service. This connection forms the tunnel. This tunnel is secured by asymmetric encryption. The private key is shared with the agent over cloud specific initialization mechanisms, such as a user-data script in AWS.

When any service on the CDP Control Plane wants to send a request to a service deployed on the customer environment (depicted in the below diagram as the “logical flow”), it physically sends a request to the CCM service running in the CDP Control Plane. The CCM agent and CCM service collaborate over the established tunnel to accept the request, forward it to the appropriate service, and send a response over the tunnel to be handed over the calling service on the CDP Control Plane.


CLOUDERA CDP CONTROL PLANE
CUSTOMER ENVIRONMENT
Sevice on Control Plane
Services on Customer Environment
Logical Flow: Application Level. HTTPS
Physical Flow: CCM Protocol (SSH/HTTPS)
CCM Service
LB
CDP Control Plane
CCM Agent

Currently, all EKS clusters provisioned by various CDP data services are enabled with public and private cluster endpoints even under fully private network setup (see Amazon EKS cluster endpoint access control). The EKS public endpoint is needed to facilitate the interactions between CDP Control Plane and the EKS cluster while worker nodes and Kubernetes control plane interact over private API endpoints. There are plans to support private EKS endpoints in the future. When this occurs, the documentation will be updated to reflect the same.

Fully private outbound restricted networks

Fully private outbound restricted networks is a variant of the fully private network where customers would like to pass outbound traffic originating from their cloud account through a proxy or firewall and explicitly allow-list URLs that are allowed to pass through. CDP Public Cloud supports such configuration. If suvh network architecture is chosen, the customer must ensure the following:

- Users configure a proxy for the environment via CDP, as documented in Use a non-transparent proxy with Cloudera Data Warehouse on AWS environments for Cloudera Data Warehouse and Using a non-transparent proxy for all other compute workloads and the Data Lake itself.
- Compute resources (such as VMs used by Data Hubs and data services) can connect to the proxy or firewall via appropriate routing rules.
- The proxy or firewall is set up to allow connections to all hosts, IP ranges, ports, and protocol types that are documented in Outbound network access destinations for AWS.

Note:

Given that fully private networks is the recommended option of connectivity in most cases, this document describes the architecture assuming a fully private network setup.

Customer on-prem network to cloud network

After compute workload clusters are launched in the customer's cloud network, data consumers such as data engineers, data scientists, and data analysts access services running in these CDP data services. Sometimes, CDP admins who set up and operate these clusters might need this access to diagnose any issues the clusters face.

Examples of these include:

- Web UIs such as:
  - Hue: For running SQL queries in Hive tables
  - CML Workspaces: For accessing Cloudera Machine Learning projects, models, notebooks, and so on
  - Cloudera Manager: For Data Hubs and Data Lakes
  - Atlas and Ranger: For metadata, governance, and security in the Data Lake
- JDBC endpoints: Customers can connect tools such as Tableau using a JDBC URL pointing to the Hive server.
- SSH access: Data engineers might log in to nodes on the compute CDP data services to run data processing jobs using YARN, Spark, or other data pipeline tools.
- Kube API access: CDP data services that run on Amazon EKS (such as Cloudera Data Warehouse and Cloudera Machine Learning) also provide admin access to Kubernetes for purposes of diagnosing issues.
- API access: Customers can use APIs for accessing many of the services exposed via the web UIs for purposes of automation and integration with other tools, applications, or other workloads they have. For example, CML exposes the "CML API v2" to work with Cloudera Machine Learning projects and other entities. See CML API v2.

These services are accessed by these consumers from within a corporate network inside a VPN. These services typically have endpoints that have a DNS name, the format of which is described more completely in the DNS section of this reference architecture documentation. These DNS names resolve to IP addresses assigned to the nodes, or load balancers fronting the ingest controllers of Kubernetes clusters. Note that these IP addresses are usually private IPs; Therefore, in order to be able to connect to these IPs from the on-premise network within a VPN, some special connectivity setup would be needed, typically accomplished using technologies like VPN peering, DirectConnect, transit gateways, and so on. While there are many options possible here, this document describes one concrete option of achieving this connectivity.

Network architecture

Cloudera recommends that customers configure their cloud networks as fully private networks, as described in this chapter. This will help on-boarding CDP Data Lakes, Data Hubs, and data services smoothly.

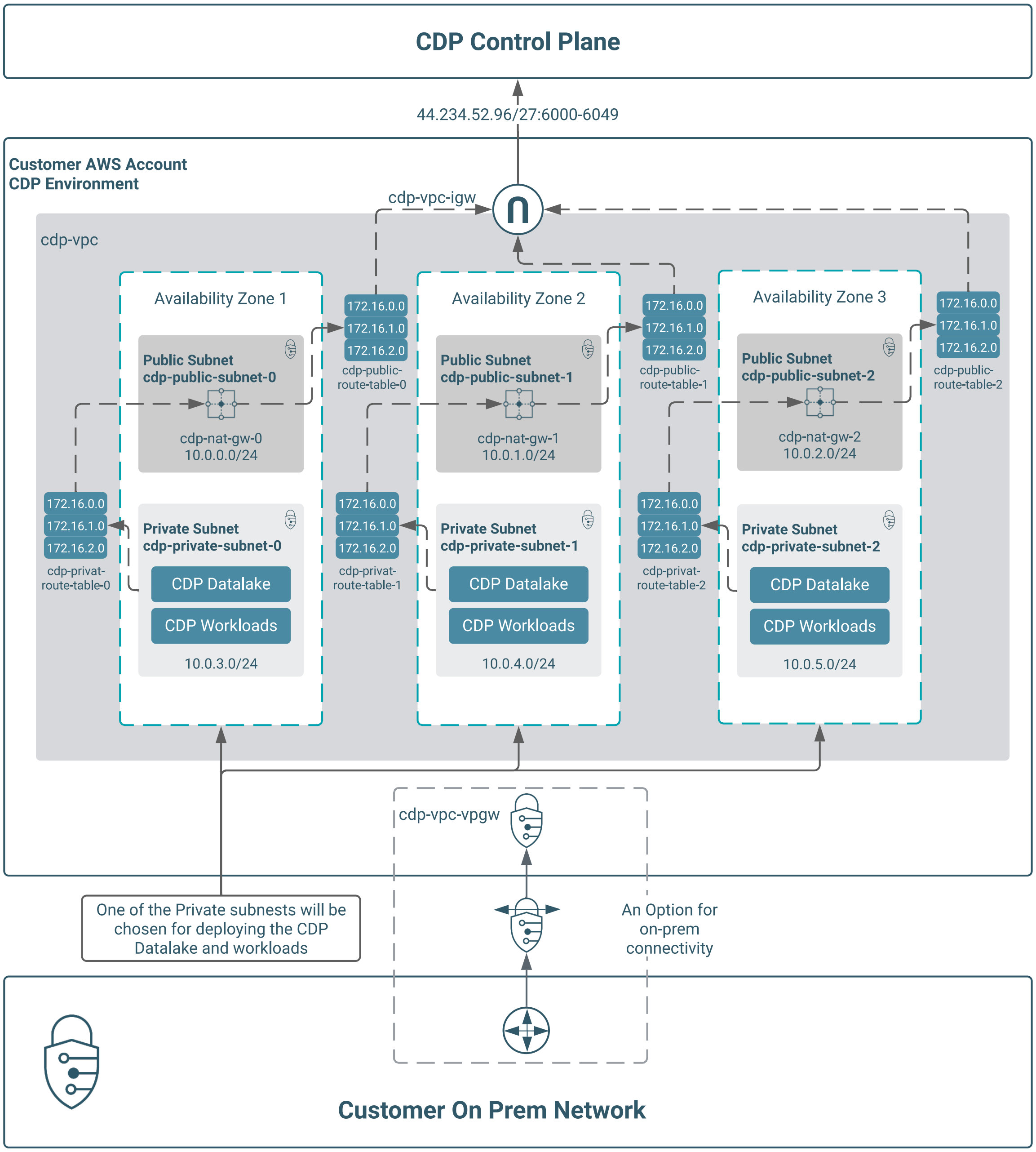
Note: This network architecture only covers the fully private networks and assumes unrestricted outbound access.

The cdpctl tool, which is released along with this document can be used to automatically set up a model of this reference architecture, which can then be reviewed for security and compliance purposes.

Architecture diagrams

This topic includes diagrams illustrating the various elements of the network architecture in the customer's cloud account into which CDP environments with Data Lakes, Data Hubs, and data services will be launched.

Cloudera recommends that customers configure their cloud networks as described in this chapter and illustrated in the following diagrams. This will help onboarding Data Lakes, Data Hubs, and data services smoothly. The following diagram illustrates the configuration for a fully private network that can be configured by the customer. This configuration can be provided by the CDP admins when they are setting up CDP environments and workloads which will get launched into this configuration.



Note the following points about the architecture:

- The configuration is a fully private network configuration - that is, the workloads are launched on nodes that do not have public IP addresses into a private subnet.
- They connect outbound to the CDP Control Plane over a fixed IP and port range.
- For users to be able to connect from the customer on-prem network to the CDP workloads in the private subnet, some network connectivity setup is required. In this case, a customer's VPN server peered to an AWS virtual private gateway is shown.

Some of the CDP data services are based on AWS EKS clusters. Amazon EKS manages the Kubernetes Control Plane while the worker nodes that make up the cluster get provisioned in the customer's VPC. The EKS Control Plane has an API endpoint for administrative purposes which is commonly referred to as "cluster endpoint". The CDP data service itself is accessible through a service endpoint ELB.

This is illustrated in the following diagram:

As can be seen in the above diagram, CDP workloads have dependencies on some AWS cloud services such as RDS, EFS and so on. A full list of these services, described in the context of each workload is specified in AWS outbound network access destinations.

In the chapters that follow, we detail the elements of this architecture, including specifying the configuration and options in each of the components.

Component description

This topic provides an overview of the VPC, subnets, gateways and route tables, and security groups required for CDP Public Cloud for AWS.

VPC

An Amazon Virtual Private Cloud (VPC) is needed for deploying CDP workloads into the customer's cloud account. Cloudera recommend that the VPC used for CDP is configured with properties as specified in this topic.

- The CIDR block for the VPC should be sufficiently large for supporting all the Data Hubs and data services that you intend to run. Refer to Determining the CIDR range for understanding how to compute the CIDR block range.
- The VPC properties for DNS hostnames and DNS resolution must be ENABLED. DNS resolution lets Kubernetes pods resolve external host names and also to support DNS hostnames. The DNS hostnames option needs to be enabled as several CDP data services rely on EFS (see Mounting on Amazon EC2 with a DNS name). Enabling these properties is also a requirement (see Amazon EKS cluster endpoint access control) to enable private access of EKS cluster endpoints.
- VPCs are associated with a DHCP Option Set. The DHCP option set for the VPCs must be set up as per the section described in DHCP option set.

Subnets

A subnet is a partition of the virtual network in which CDP workloads are launched.

It is recommended that the subnets be configured with the following properties:

- It is recommended to have 3 private subnets and 3 public subnets, such that each private-public subnet pair is in a different availability zone (AZ). Even if a region has two AZs instead of three, it's recommended that three private subnets are created, two in the same AZ. This is required to prevent cross AZ routing of traffic and to maintain Quorum-based consistency required by some services.
  - Note that a subnet becomes 'private' or 'public' based on the routing devices it is associated with in the route tables. This is described in Gateways and route tables.
  - The private subnets will be where the compute workloads will be launched by CDP. This ensures that these nodes are working in an isolated and secure environment that does not have internet connectivity.
  - The public subnet is needed to host a NAT gateway as this will allow the compute nodes to reach out to the CDP Control Plane over the internet. More on this will be described in Gateways and route tables.
- The CIDR block for the subnets should be sufficiently large for supporting all the CDP data services you intend to run. Refer to Determining the CIDR range for understanding how to compute the CIDR block range.
- The CIDR block for the subnets should not overlap with known AWS EKS ranges for pods/services. Several EKS based CDP data services in Overlay networks
- In addition, you may want to ensure that the CIDR ranges assigned to the Subnets will not overlap with any of your on-premise network CIDR ranges, as this may be a requirement for setting up connectivity from your on-prem network to the subnets.
- Since Cloudera recommends fully private network configuration, the 'Auto-assign public IPs' option must be disabled for the private subnets.
- A subnet can be associated with a Network ACL (NACL). However, since Cloudera works with a fully private network configuration where communication is always initiated from EC2 nodes within the subnets, a NACL is generally not useful for this configuration.
- Tag private subnets with a tag 'kubernetes.io/role/internal-elb:1'. The key is the string and the value is '1'. Cloud Controller Manager and AWS Load Balancer Controller both require private subnets to have this tag for automatic creation of private ELBs. Private ELBs are created in these subnets by EKS. This is applicable when CDP is supporting EKS versions < 1.20 (which is currently the case). See How can I tag the Amazon VPC subnets in my Amazon EKS cluster.

Gateways and route tables

This topic covers recommended gateway and route table configurations for CDP Public Cloud for AWS.

Connectivity from Control Plane to CDP workloads

- As described in Taxonomy of network architectures, nodes in the CDP workloads need to connect to the CDP Control Plane over the internet to establish a 'tunnel' over which the CDP Control Plane can send instructions to the workloads.

- In order to accomplish this, there are two gateways that need to be configured - a NAT Gateway in each of the public subnets and an Internet Gateway at the VPC level.
- The private subnet hosting the CDP workloads should be configured with a route table where the default route (0.0.0.0/0) points to a NAT Gateway in the public subnet of its AZ.
- The public subnet hosting the NAT Gateway should be configured with a route table where the default route (0.0.0.0/0) points to an Internet Gateway the VPC is configured with.
- Each NAT gateway requires an elastic IP address. The VPC should contain as many elastic IP addresses as NAT gateways across the AZs in the VPC.

Connectivity from customer on-prem to CDP workloads

- As described in Use cases, data consumers need to access data processing or consumption services in the CDP workloads. Given these are created with private IP addresses in private subnets, the customers will need to arrange for access to these addresses from their on-prem or corporate networks in specific ways.
- There are several possible solutions for achieving this, but one that is depicted in the Architecture diagram, uses a AWS VPN Gateway service.
- In this solution, the customer has to create a Virtual Private Gateway, and connect it to the VPN service on the on-prem network.

Security groups

During the specification of a VPC to CDP, the CDP admin specifies the security groups that will be associated with all the CDP workloads launched within that VPC. These security groups will be used in allowing the incoming traffic to the hosts.

Security groups for Data Lakes and Data Hubs

During the specification of a VPC to CDP, the CDP admin can either let CDP create security groups, taking a list of IP address CIDRs as input; or create them in AWS and then provide them to CDP.

When getting started with CDP, the CDP admin can let CDP create security groups, taking a list of IP address CIDRs as input. These will be used in allowing the incoming traffic to the hosts. The list of CIDR ranges should correspond to the address ranges from which the CDP workloads will be accessed. In a VPN-peered VPC, this would also include address ranges from customer’s on-prem network. This model is useful for initial testing given the ease of set up.

Alternatively, the CDP admin can create security groups on their own and select them during the setup of the VPC and other network configuration. This model is better for production workloads, as it allows for greater control in the hands of the CDP admin. However, note that the CDP admin must ensure that the rules meet the requirements described below.

For a fully private network, security groups should be configured according to the types of access requirements needed by the different services in the workloads:

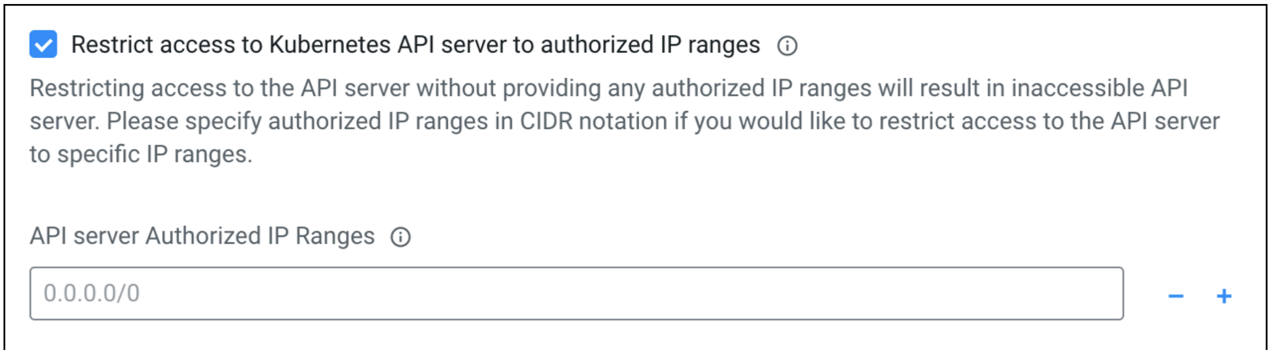
- Services accessed only within the VPC must be configured with the following inbound rules:
  - All TCP / UDP / ICMP access is allowed for the CIDRs corresponding to the VPC.
  - Conversely, there is no need to provide any access to these services for any IP CIDRs outside the VPC.
- Endpoint services are the services that can be accessed outside the VPC through the gateway, chiefly by data consumers or CDP admins. For example, UIs like Hue, Atlas, Ranger, Cloudera Manager all need to be accessed by data consumers or other administrators. For enabling this, the following in-bound rules are set up:
  - All TCP / UDP / ICMP access is allowed for the CIDRs corresponding to the VPC.
  - All TCP ports that correspond to services like Kafka, HBase, and so on that need to be accessed outside the VPC are to be allowed for the list of CIDR ranges specified at the time of creating the environment/CDP data service, or in the security group created by the CDP admin. Alternatively, all TCP / UDP / ICMP access may be allowed for these CIDR ranges.
  - SSH access is allowed for the CIDR ranges specified at the time of creating the environment.
  - HTTPS access is allowed for the CIDR ranges specified at the time of creating the environment.

- Note that for a fully private network, even specifying an open access here (such as 0.0.0.0/0) is restrictive because these services are deployed in a private subnet without a public IP address and hence do not have a route to the Internet gateway. However, the list of CIDR ranges may be useful to restrict which private subnets of the customer's on-prem network can access the services. Rules for EKS based workloads are described in the following section.
- Rules for AKS based workloads are described separately in the following section.

Additional rules for EKS-based workloads

At the time of enabling a CDP data service, the CDP admin can specify a list of CIDR ranges that will be used in allowing the incoming traffic to the workload Elastic Load Balancer (ELB). This list of CIDR ranges should correspond to the address ranges from which the CDP data service workloads will be accessed. In a VPN peered VPC, this would also include address ranges from customer's on-prem network. In a fully private network setup, 0.0.0.0/0 implies access only within the VPC and the peered VPN network which is still restrictive.

Since a public endpoint is enabled by default for all EKS cluster Control Planes at the moment, it is highly recommended to provide a list of outbound public CIDR ranges at the time of provisioning a CDP data service to restrict access to the EKS clusters. By default, the public endpoint is always allowed to connect to the CDP public CIDR range. The following screenshot is an example configuration section for a CDP data service:



☑ Restrict access to Kubernetes API server to authorized IP ranges ⓘ

Restricting access to the API server without providing any authorized IP ranges will result in inaccessible API server. Please specify authorized IP ranges in CIDR notation if you would like to restrict access to the API server to specific IP ranges.

API server Authorized IP Ranges ⓘ

0.0.0.0/0

Specific guidelines for restricting access to Kubernetes API server and workloads are detailed in Restricting access for CDP services that create their own security groups on AWS by each CDP data service.

Within the EKS cluster, there are several security groups defined to facilitate EKS control plane-pod communication, inter-pod and inter-worker node communication as well as workload communication through ELBs. These groups are in accordance with AWS documentation (see Amazon EKS security group considerations.)

Outbound connectivity requirements

Outbound traffic from the worker nodes is unrestricted and is targeted at other AWS services and CDP services. The comprehensive list of services that get accessed from a CDP environment can be found in AWS documentation (see Amazon EKS security group considerations).

DNS

This topic covers recommended DNS configurations for CDP Public Cloud for AWS.

The previous sections dealt with how connectivity is established to the workload infrastructure. This section deals with 'addressability'. The workloads launched by CDP contain a few services that need to be accessed by the CDP admins or data consumers. These include services like Cloudera Manager, metadata services like the Hive Metastore, Atlas or Ranger, data processing or consumption services such as Oozie server, Hue, and so on. Given the nature of the cloud infrastructure, the IP addresses for the nodes running these services may change (for example, if the infrastructure is restarted or repaired). However, these should have statically addressable DNS names so that users can access them with the same names.

In order to help with this, CDP assigns DNS names to these nodes. These naming schemes have the following properties:

- The DNS name is of the following format for each Data Lake node, Data Hub node, and the Data Lake/Data Hub cluster endpoint: <CLUSTER_NAME>-{<HOST_GROUP><i>}.<ENVIRONMENT_IDENTIFIER>.<CUSTOMER_IDENTIFIER>.cloudera.site

  An example could be my-dataeng-master0.my-envir.aaaa-1234.cloudera.site

  This name has the following components:

  - The base domain is cloudera.site. This is a publicly registered DNS suffix (see Public Suffix List). It is also a registered Route53 hosted zone in a Cloudera owned AWS account.
  - The <CUSTOMER_IDENTIFIER> is unique to a customer account on CDP made of alphanumeric characters and "-"
  - The <ENVIRONMENT_IDENTIFIER> is generated based on the environment name and is truncated to 8 characters.
  - The <CLUSTER_NAME> is the cluster name given to the Data Lake or Data Hub. It is appended with a <HOST_GROUP> name such as "gateway", "master", "worker", and so on, depending on the role that the node plays in the cluster. If there are more than one of these nodes playing the same role, they are appended with a serial number, <i>.

- The DNS name of the endpoints of the CDP data services is of the following format:
    - For a Virtual Warehouse in CDW, it is <VIRTUAL_WAREHOUSE_NAME>.<CDW_ENVIRONMENT_IDENTIFIER>.dw.<CUSTOMER_IDENTIFIER>.cloudera.site
        - The <VIRTUAL_WAREHOUSE_NAME> is the name of the Virtual Warehouse created. There could be multiple virtual warehouses for a given CDP environment.
        - The <CDW_ENVIRONMENT_IDENTIFIER> is the identifier for the CDP environment.
    - For a Session Terminal in a CML workspace, it is <TTY_SESSION_ID>.<CML_WORKSPACE_ID>.<ENVIRONMENT_IDENTIFIER>.<CUSTOMER_IDENTIFIER>.cloudera.site
        - The <TTY_SESSION_ID> is the ID of the CML Terminal Session ID.
        - The <CML_WORKSPACE_ID> is the ID of the CML workspace created.
        - The <ENVIRONMENT_IDENTIFIER> is generated based on the environment name and is truncated to 8 characters. If the 8th character is a "-" (dash), then it is truncated to 7 characters instead.
    - For all the CDP data services listed above, the common portions of the DNS include.
        - The base domain is cloudera.site. This is a publicly registered DNS suffix (see Public Suffix List). It is also a registered Route53 hosted zone in a Cloudera owned AWS account.
        - The <CUSTOMER_IDENTIFIER> is unique to a customer account on CDP made of alphanumeric characters and a "-" (dash).
    - For a virtual cluster in CDE, it is <VIRTUAL_CLUSTER_ID>.<CDE_SERVICE_ID>.<ENVIRONMENT_IDENTIFIER>.<CUSTOMER_IDENTIFIER>.cloudera.site
        - The <VIRTUAL_CLUSTER_ID> is the 8-character ID of the CDE virtual cluster, for example, afg57p98.
        - The <CDE_SERVICE_ID> is the ID of the CDE service containing the virtual cluster, for example, cde-g6th4kjv.
        - The <ENVIRONMENT_IDENTIFIER> is generated based on the CDP environment name and is truncated to 8 characters. if the 8th character is a "-" (dash), then it is truncated to 7 characters instead.
    - For a DataFlow service in CDF, it is

      dfx.<CDF_WORKLOAD_ENDPOINT_ID>.<CUSTOMER_IDENTIFIER>.cloudera.site
        - The <CDF_WORKLOAD_ENDPOINT_ID> is the 8-character ID of the CDP DataFlow Service Workload Endpoint, for example, 1bxt50kk.
    - For a database in COD, it is <COD_WORKLOAD_NAME>-{<HOST-GROUP><i>}.<ENVIRONMENT_IDENTIFIER>.<CUSTOMER_IDENTIFIER>.cloudera.site
        - <COD_Workload_Name> is the ID of the Cloudera Operational Database, for example, cod-1m6yz9uwqhrg2.
        - The user provides a database name and the environment where they want to create the database. These two entities are hashed together to create the internal <COD_WORKLOAD_NAME>, which is set as the Data Hub cluster.
        - Except for the <COD_WORKLOAD_NAME>, the rest of the DNS name of the endpoint is implemented as per Data Hub DNS format as mentioned above.
    - For all the CDP data services listed above, the common portions of the DNS include:
        - The base domain is cloudera.site. This is a publicly registered DNS suffix. It is also a registered Route53 hosted zone in a Cloudera owned AWS account.
        - The <CUSTOMER_IDENTIFIER> is unique to a customer account on CDP made of alphanumeric characters and a "-" (dash).
- The length of the DNS name is restricted to 64 characters due to some limitations with Hue workloads.
- These names are stored as A records in the Route53 hosted zone in the Cloudera managed CDP Control Plane AWS account. Hence, you can resolve these names from any location outside of the VPC. However, note that they would still resolve to private IP addresses and hence are constrained by the connectivity setup described in preceding sections.
- Within a CDP environment, the DNS resolution happens differently. Every CDP environment has a DNS server that is played by a component called FreeIPA. This server is seeded with the hostnames of the nodes of all

workload clusters in the environment. Every node in a Data Lake, Data Hub, and a CDP data service is configured to look up the FreeIPA DNS service for name resolution within the cluster.

DHCP option set

This topic covers recommended DHCP Option Set configurations for CDP Public Cloud for AWS.

- Ensure the DHCP Option Set has the DNS Server set to AmazonProvidedDNS. This is required for EFS (see Mounting on Amazon EC2 with a DNS name) and EKS (see Amazon EKS cluster endpoint access control).
- Customers may choose to use their own DNS servers from their corporate network in the DHCP option set. However, the DNS servers need to be configured to conditionally forward DNS queries for the Amazon subdomains to Amazon Provided DNS resolver in the VPC. This also helps ensure the CDP connectivity to AWS services. If the CDP data services do not allow activation with the custom DNS servers, please contact Cloudera support to get the relevant entitlements added.
- Ensure the DHCP Option Set associated with the VPC has only one domain and it is recommended that it remains default.

Determining the CIDR range

This topic covers options for determining the CIDR range for CDP Public Cloud for AWS.

When registering an AWS environment in CDP, you are asked to select a VPC and one or more subnets. You have two options:

1. CDP will create a new VPC and subnets.
2. Select an existing VPC.

In both cases, use this guide to calculate and verify the limits of the VPC and subnets available in your AWS subscription to ensure that you have enough networking resources to create clusters in CDP.

Option 1: CDP creates the VPCs and subnets

If you would like CDP to create a new VPC, you will need to specify a /16 CIDR in IPv4 range that will be used to define the range of private IPs for VM instances provisioned into these subnets. The default is 10.10.0.0/16.

By default CDP creates 6 subnets (3 private and 3 public) and divides the address space as follows:

- 3 x /19 private subnets for FreeIPA, Data Lake, Data Hub, Data Warehouse, Machine Learning, Data Engineering, DataFlow, Operational Database.
- 3 x /24 public subnets reserved for future use.
- CDP creates an internet gateway for your VPC and deploys a NAT gateway in each public subnet and configures it in the route table of each private subnet.
- All the resources created in the VPC are assigned with relevant tags (like ‘kubernetes.io/role/internal-elb:1’ for private subnets and ‘kubernetes.io/role/elb:1’ for public subnets).

Option 2: Existing VPC and subnets

If you would like to use an existing VPC, the subnet requirements vary based on the services used. This section is a guide for calculating network requirements per service.

Subnets for Data Lake and Data Hub

Both Data Lake and Data Hub share the same subnet, but due to Data Lake’s dependency on RDS, CDP admins need at least two subnets.

It is recommended to use subnets of /19 CIDR. If you would like to use a smaller subnets, use the following guidelines:

- One IP address is used for each VM.

- Two IPs for your RDS instances.
- One Light Duty Data Lake cluster uses 2 VMs and 1 additional VMs for FreeIPA.
- One Medium Duty Data Lake cluster uses 10 VMs and 3 additional VMs for FreeIPA.
- The exact number of VMs depends on the Data Hub's cluster definition, but a typical Data Hub cluster uses a minimum of four VMs as a starting point and this number can be dynamically scaled up or down.
- Make sure you allocate enough IPs to handle each cluster running at peak capacity

Subnets for Data Warehouse

The Data Warehouse service needs three subnets. You can choose the specific subnets that should be used when you activate Data Warehouse for an environment and if the specified subnets are shared with other CDP data services then ensure that there are sufficient IPs left for CDW.

An environment can be activated for CDW with Overlay Network or without Overlay Network, the next section outlines the benefits and the pitfalls of one choice over the other.

Using Data Warehouse with Overlay Network vs without Overlay Network

Enabling overlay networks creates two network spaces in your EKS:

- A node network space, which derives per-node IP addresses from the VPC.
- A Kubernetes pod network space, which derives per-pod IP addresses from the CNI plugin's own network space.

The overlay network is bridged into the node network. As a result, one IP address is required per node instead of one IP address needed per pod. It is recommended to enable overlay networks with Data Warehouse if the available IPs in your subnets is less than 1024.

Even though using an overlay network for CDW requires less number of IPs it adds more metadata to each network packet which even though are processed in the kernel can have a performance impact. Due to this extra metadata added to each network packet it may not be straightforward to debug incoming and outgoing traffic with some of the network debugging tools. To employ an overlay network for CDW, Weave net CNI (Container Network Interface) plugin is used. Since weave-net is a full-mesh network where each node is connected to every other node in the mesh, the number of nodes the overlay network can span across is limited to 200.

Data Warehouse with Overlay Networks

Use these guidelines to arrive at your desired subnet CIDR if you choose to use overlay networks. It is recommended to enable overlay networks with Data Warehouse if the available IPs in your subnets is less than 1024.

VM type	No of VMs	Total no of IPs addresses required
DW Shared Services - (Shared among all VWs in an environment)	3	3
Per Database Catalog (One catalog is created by default, you can create additional catalogs)	2	2
Shared Services per Virtual Warehouse (HS2, Hue, DAS, coordinators, catalog, statestored, and so on.)	3	3
Per Virtual Warehouse (XS) *	2 to 20	2 to 20
Per Virtual Warehouse (S) *	10 to 100	10 to 100
Per Virtual Warehouse (M) *	20 to 200	20 to 200
Per Virtual Warehouse (L) *	40 to 400	40 to 400
Per Virtual Warehouse (Custom)*	x to 10x (where x is the initial node count)	x to 10x (where x is the initial node count)

* Each autoscaling activity can be treated as deploying a new Virtual Warehouse. For example, when a XS Virtual Warehouse is scaled once, it uses four VMs instead of two.

Data Warehouse without Overlay Network

Use these guidelines to arrive at your desired subnet CIDR if you choose to not to use overlay networks while activating your DW environment.

VM type	No of VMs	Maximum no of pods per VM	Maximum number of IPs per VM (No of pods per VM +1 per VM)	Maximum no of IPs addresses required
DW Shared Services - (Shared among all VWs in an environment)	3	25	26	78
Per Database Catalog (One catalog is created by default, you can create additional catalogs)	2	25	26	52
Shared Services per Virtual Warehouse (HS2, Hue, DAS, coordinators, catalog, statestored, and so on.)	3	25	26	78
Per Virtual Warehouse (XS) *	2 to 20	10	11	22 to 220
Per Virtual Warehouse (S) *	10 to 100	10	11	110 to 1100
Per Virtual Warehouse (M) *	20 to 200	10	11	220 to 2200
Per Virtual Warehouse (L) *	40 to 400	10	11	440 to 4400
Per Virtual Warehouse (Custom)*	x to 10 x (where x is the initial node count)	10	11	11x to 110x

* Each autoscaling activity can be treated as deploying a new Virtual Warehouse. For example, when a XS Virtual Warehouse is scaled once, it uses four VMs instead of two.

Query Isolation

If the Query Isolation feature has been enabled for a Virtual Warehouse and a query scans more than the threshold set in the hive.query.isolation.scan.size.threshold parameter, the planner runs the query in isolation. This means that an isolated standalone executor group is spawned to run the data-intensive query. The number of executors spawned to run the query does not exceed the default setting for the hive.query.isolation.max.nodes.per.query parameter, which default to 2 times the virtual warehouse template size. The number of isolated parallel queries (can go up to 400) and number of nodes per isolated queries (can go up to 400) is configurable for a given virtual warehouse. Each of these nodes can consume up to 11 IPs when overlay networks are not enabled and 1 IP if the overlay network is enabled against your environment.

Subnets for Machine Learning

CML requires at least two subnets in two different availability zones and you can choose which subnets should be used by a workspace at the time of provisioning. If the specified subnets are shared with other CDP data services then ensure that there are sufficient IPs left for CML. CML uses Calico CNI (Container Network Interface) to run ML pods in an overlay network. The formula to calculate IP Addresses per workspace is as follows:

- Each workspace can grow up to 100 CPU worker nodes and 100 GPU workers; each node consumes 1 IP address.
- In addition, you will need to allocate 9 IP addresses for infrastructure nodes (3 IPs for ML infra nodes, 4 for Liftie infra nodes and 2 load balancers).

Subnets for Data Engineering

CDE requires at least two subnets in two different availability zones. You choose at the time of workspace provisioning which subnets should be used. Ensure that there are adequate IPs left for CDE to provide for the maximum expected size of the cluster. CDE uses Calico CNI (Container Network Interface) to run pods in an overlay network. A /24 CIDR is recommended for CDE subnets, but for a custom range the formula to calculate IP addresses per CDE service is as follows:

- Each CDE service can scale up to 100 compute nodes, each node consumes one IP address.
- In addition, you need to allocate 5 IP addresses for the infrastructure nodes (1 IP for DE infra node, 4 for Liftie infra nodes) and 2 IP addresses per virtual cluster for the virtual cluster service nodes.

Subnets for DataFlow

CDF requires at least two subnets in two different availability zones. DataFlow by default configures EKS to run in private subnets, if they are available. CDF uses Calico CNI (Container Network Interface) to run pods in an overlay network. The CIDR block for the subnets must be sized appropriately in each CDF environment to accommodate the following:

- Each DataFlow cluster can grow up to 50 autoscaling compute instances, each of which consumes 1 IP address.
- A fixed overhead of 48 IP addresses for three instances for core DataFlow services.

Subnets for Operational Database

COD currently leverages Data Hub to deploy infrastructure in a private subnet. Clients associated with HBase REST server, Thrift Server, or the Phoenix Thin server can be proxied via the VPN gateway. If Apache HBase Java API or Apache Phoenix Thick JDBC client are used, an edge node must be configured to access the private computing resources. The formula to calculate IP addresses per COD database is as follows:

- Each COD database defaults to a minimum of 9 nodes (1 leader, 2 master, 1 gateway, and 5 worker nodes), requiring 9 IP addresses.
- Each COD database can autoscale the number of worker nodes given various factors/attributes. The range defaults to a minimum of 5 nodes and a maximum of 20 nodes. Users can reduce the minimum to 3 nodes, but there is currently no fixed upper limit. They must consider the behaviour of their database while providing the CIDR range to accommodate for the potential autoscaling growth, where each node takes 1 IP address.
- In addition, configuring an edge node, if required for client applications, takes 1 IP address for 1 accessory node.

DNS

This topic covers recommended DNS configurations for CDP Public Cloud for AWS.

The previous sections dealt with how connectivity is established to the workload infrastructure. This section deals with ‘addressability’. The workloads launched by CDP contain a few services that need to be accessed by the CDP admins or data consumers. These include services like Cloudera Manager, metadata services like the Hive Metastore, Atlas or Ranger, data processing or consumption services such as Oozie server, Hue, and so on. Given the nature of the cloud infrastructure, the IP addresses for the nodes running these services may change (for example, if the infrastructure is restarted or repaired). However, these should have statically addressable DNS names so that users can access them with the same names.

In order to help with this, CDP assigns DNS names to these nodes. These naming schemes have the following properties:

- The DNS name is of the following format for each Data Lake node, Data Hub node, and the Data Lake/Data Hub cluster endpoint: <CLUSTER_NAME>-{<HOST_GROUP><i>}.<ENVIRONMENT_IDENTIFIER>.<CUSTOMER_IDENTIFIER>.cloudera.site

  An example could be my-dataeng-master0.my-envir.aaaa-1234.cloudera.site

  This name has the following components:

  - The base domain is cloudera.site. This is a publicly registered DNS suffix (see Public Suffix List). It is also a registered Route53 hosted zone in a Cloudera owned AWS account.
  - The <CUSTOMER_IDENTIFIER> is unique to a customer account on CDP made of alphanumeric characters and "-"
  - The <ENVIRONMENT_IDENTIFIER> is generated based on the environment name and is truncated to 8 characters.
  - The <CLUSTER_NAME> is the cluster name given to the Data Lake or Data Hub. It is appended with a <HOST_GROUP> name such as "gateway", "master", "worker", and so on, depending on the role that the node plays in the cluster. If there are more than one of these nodes playing the same role, they are appended with a serial number, <i>.

- The DNS name of the endpoints of the CDP data services is of the following format:
    - For a Virtual Warehouse in CDW, it is <VIRTUAL_WAREHOUSE_NAME>.<CDW_ENVIRONMENT_IDENTIFIER>.dw.<CUSTOMER_IDENTIFIER>.cloudera.site
        - The <VIRTUAL_WAREHOUSE_NAME> is the name of the Virtual Warehouse created. There could be multiple virtual warehouses for a given CDP environment.
        - The <CDW_ENVIRONMENT_IDENTIFIER> is the identifier for the CDP environment.
    - For a Session Terminal in a CML workspace, it is <TTY_SESSION_ID>.<CML_WORKSPACE_ID>.<ENVIRONMENT_IDENTIFIER>.<CUSTOMER_IDENTIFIER>.cloudera.site
        - The <TTY_SESSION_ID> is the ID of the CML Terminal Session ID.
        - The <CML_WORKSPACE_ID> is the ID of the CML workspace created.
        - The <ENVIRONMENT_IDENTIFIER> is generated based on the environment name and is truncated to 8 characters. If the 8th character is a "-" (dash), then it is truncated to 7 characters instead.
    - For all the CDP data services listed above, the common portions of the DNS include.
        - The base domain is cloudera.site. This is a publicly registered DNS suffix (see Public Suffix List). It is also a registered Route53 hosted zone in a Cloudera owned AWS account.
        - The <CUSTOMER_IDENTIFIER> is unique to a customer account on CDP made of alphanumeric characters and a "-" (dash).
    - For a virtual cluster in CDE, it is <VIRTUAL_CLUSTER_ID>.<CDE_SERVICE_ID>.<ENVIRONMENT_IDENTIFIER>.<CUSTOMER_IDENTIFIER>.cloudera.site
        - The <VIRTUAL_CLUSTER_ID> is the 8-character ID of the CDE virtual cluster, for example, afg57p98.
        - The <CDE_SERVICE_ID> is the ID of the CDE service containing the virtual cluster, for example, cde-g6th4kjv.
        - The <ENVIRONMENT_IDENTIFIER> is generated based on the CDP environment name and is truncated to 8 characters. if the 8th character is a "-" (dash), then it is truncated to 7 characters instead.
    - For a DataFlow service in CDF, it is

      dfx.<CDF_WORKLOAD_ENDPOINT_ID>.<CUSTOMER_IDENTIFIER>.cloudera.site
        - The <CDF_WORKLOAD_ENDPOINT_ID> is the 8-character ID of the CDP DataFlow Service Workload Endpoint, for example, 1bxt50kk.
    - For a database in COD, it is <COD_WORKLOAD_NAME>-{<HOST-GROUP><i>}.<ENVIRONMENT_IDENTIFIER>.<CUSTOMER_IDENTIFIER>.cloudera.site
        - <COD_Workload_Name> is the ID of the Cloudera Operational Database, for example, cod-1m6yz9uwqhrg2.
        - The user provides a database name and the environment where they want to create the database. These two entities are hashed together to create the internal <COD_WORKLOAD_NAME>, which is set as the Data Hub cluster.
        - Except for the <COD_WORKLOAD_NAME>, the rest of the DNS name of the endpoint is implemented as per Data Hub DNS format as mentioned above.
    - For all the CDP data services listed above, the common portions of the DNS include:
        - The base domain is cloudera.site. This is a publicly registered DNS suffix. It is also a registered Route53 hosted zone in a Cloudera owned AWS account.
        - The <CUSTOMER_IDENTIFIER> is unique to a customer account on CDP made of alphanumeric characters and a "-" (dash).
- The length of the DNS name is restricted to 64 characters due to some limitations with Hue workloads.
- These names are stored as A records in the Route53 hosted zone in the Cloudera managed CDP Control Plane AWS account. Hence, you can resolve these names from any location outside of the VPC. However, note that they would still resolve to private IP addresses and hence are constrained by the connectivity setup described in preceding sections.
- Within a CDP environment, the DNS resolution happens differently. Every CDP environment has a DNS server that is played by a component called FreeIPA. This server is seeded with the hostnames of the nodes of all

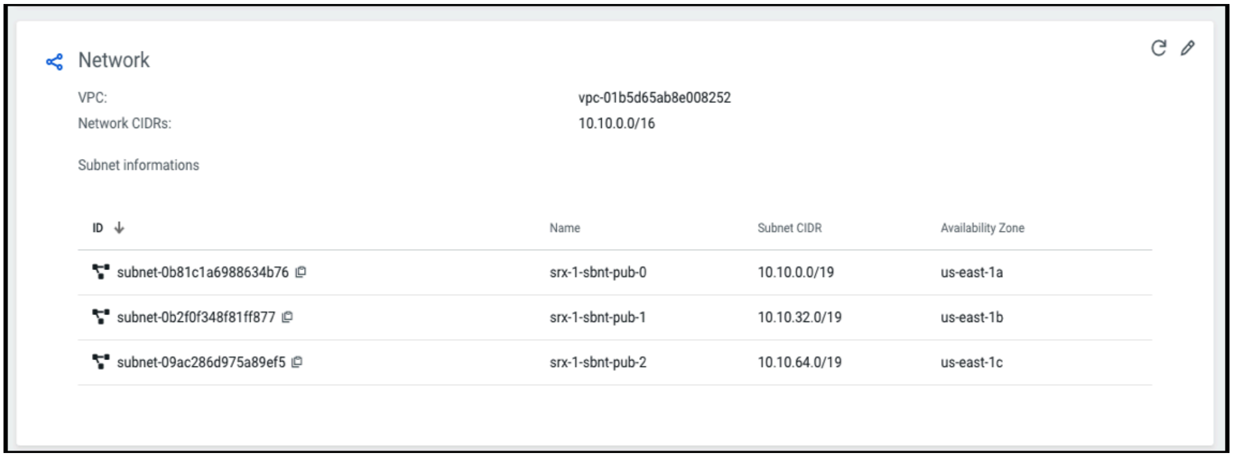
workload clusters in the environment. Every node in a Data Lake, Data Hub, and a CDP data service is configured to look up the FreeIPA DNS service for name resolution within the cluster.

Associating additional CIDRs to a VPC

You can choose to add another CIDR to your VPC in case you are close to exhausting the available IPs in your VPC by following the necessary steps from Associate a secondary IPv4 CIDR block with your VPC. Once a new CIDR has been added to your VPC, a created environment not will pick these changes automatically. For the environment to pick up the new CIDR, follow these steps.

Procedure

1. Go to your environment page and under Summary tab click on the refresh icon in the Network card.



Network

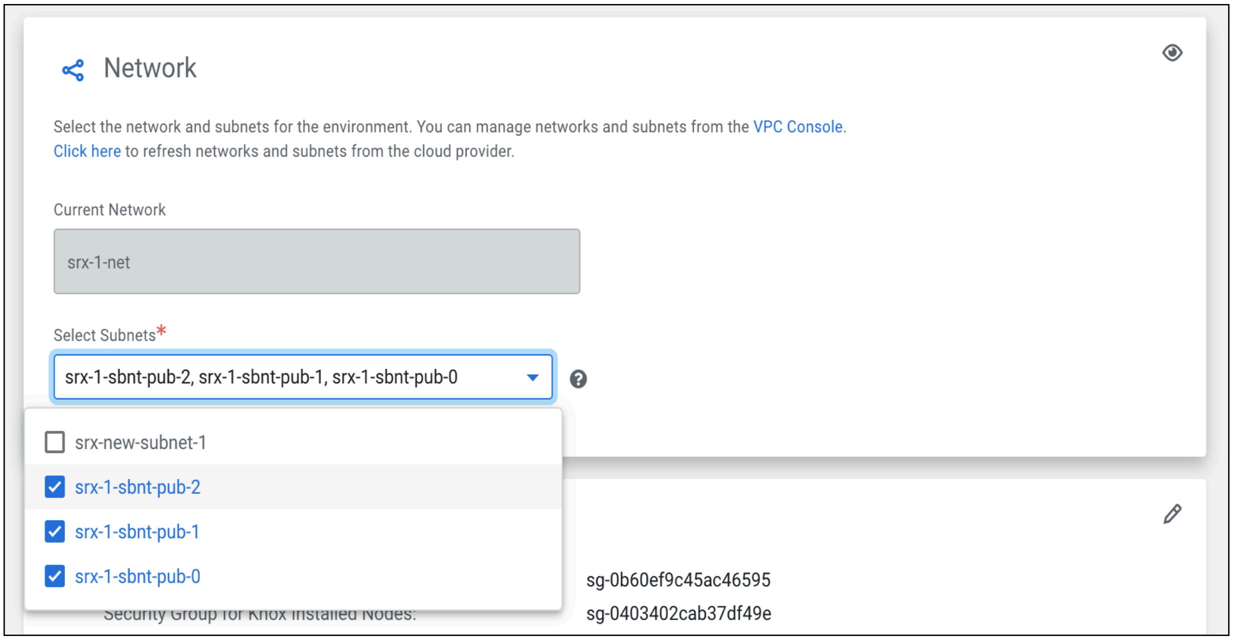
VPC: vpc-01b5d65ab8e008252

Network CIDRs: 10.10.0.0/16

Subnet informations

ID ↓	Name	Subnet CIDR	Availability Zone
subnet-0b81c1a6988634b76	srx-1-sbnt-pub-0	10.10.0.0/19	us-east-1a
subnet-0b2f0f348f81ff877	srx-1-sbnt-pub-1	10.10.32.0/19	us-east-1b
subnet-09ac286d975a89ef5	srx-1-sbnt-pub-2	10.10.64.0/19	us-east-1c

2. To add any new subnets created with the new CIDR, click on the pencil icon and choose the new subnet from the dropdown.



Network

Select the network and subnets for the environment. You can manage networks and subnets from the VPC Console.
Click here to refresh networks and subnets from the cloud provider.

Current Network

srx-1-net

Select Subnets*

srx-1-sbnt-pub-2, srx-1-sbnt-pub-1, srx-1-sbnt-pub-0

☐ srx-new-subnet-1
☑ srx-1-sbnt-pub-2
☑ srx-1-sbnt-pub-1
☑ srx-1-sbnt-pub-0

sg-0b60ef9c45ac46595

Security Group for Knox Installed Nodes: sg-0403402cab37df49e

Results

The user can now use the new subnet for any subsequent workloads that will be created.