# Cloud hub deployment

**CloudHub:** CloudHub is an Integration Platform as a Service (iPaaS). It enables you to deploy and run the application in the cloud via Runtime Manager. CloudHub is a scalable, multi-tenant, elastic, secure, and highly available iPaas. CloudHub is managed via the Runtime Manager console in the Anypoint platform. It enables you to deploy Mule applications from Anypoint Studio using the CloudHub API or the CloudHub Command Line Interface..

The Anypoint Platform **runtime plane** is where applications are deployed, and also where the Mule **runtime** engine and other application-related services, such as Anypoint Connectors run. The runtime engine includes Anypoint Security edge policies and tokenization, MQ, Object Store, and Connectors. Data and payloads are processed in runtime plane

Types of runtime plane

* CloudHub
* Runtime fabric
* Standalone runtime

The Anypoint Platform **control plane** provides a set of cloud services that simplify the design, reuse, and management of integrations and APIs. Control plane store information such as

* - metadata
* - API analytics: how many time API is called, time to took to complete a request
* - JVM utilization
* - Jar files
* - Logs

The control plane option:

1. Mulesoft hosted control plane: located in US and EU
2. Customer hosted control plane
3. Mulesoft government cloud

## Deployment options

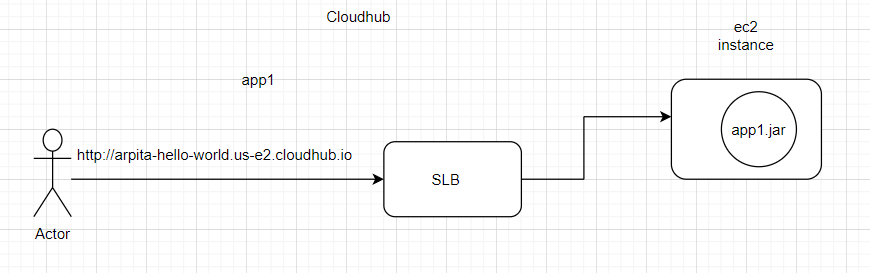
|  |  |  |
| --- | --- | --- |
|  | MuleSoft Runtime Plane | Customer Hosted Runtime Plane |
| MuleSoft Control Plane | Cloud Hub  VPC | Runtime Fabric  Hybrid |
| Customer Hosted Control Plane |  | PCE  PCF |

## Cloudhub Deployment

Cloud Hub is an iPaaS.

When we deploy an application in a cloud hub, it creates an instances called ec2 where mule jar is deployed. Each application deployed in the cloud hub are isolated from one another. It means each application have its own runtime.

To managed the ec2 instances there is a shared load balancer



* **Integration Applications**: Integration Applications are mule applications that are deployed and run in cloud
* **Runtime Manager**: Runtime Manager enables us to deploy and monitor mule applications
* **Platform Services:** This set of shared cloudhub platform services includes cloudhub insights,alerting,logging,virtual private cloud
* **Worker Cloud**: Applications on cloudhub are run by one or more instances of mule called workers

Different methods to deploy an applition in cloud hub

* runtime manager
* anypoint studio
* cloudhub CLI

Advantage of using Cloud Hub:

* No need to spend on infrastructure
* High availability
* Vertical and horizontal scaling
* Inbuild load balancing
* Provides application isolation
* Zero downtime deployment: Zero downtime deployment is a deployment method where your website or application is never down or in an unstable state during the deployment process. To achieve this the web server doesn't start serving the changed code until the entire deployment process is complete
  + Get continuous software updates, no complex hardware maintenance required. Fully Managed
  + Improve efficiency with multi-tenancy for applications and workers
  + Enjoy a globally distributed architecture with 99.99% uptime. Globally available
  + Leverage out-of-the-box cloud security and compliance
  + Control access based on complex organizational requirements

Q. When does customer choose cloud hub deployment?

* When the customer do not have a proper infrastructure to deploy runtime customer prefer to use cloud hub deployment. In this everything is managed by the mulesoft

Ec2 – elastic cloud instance

* Applications are deployed here
* AWS offers a wide range of memory optimized instances ranging from 244 GB R4 instances to 24 TB High Memory instances.

Q. Why multiple mule instances are not allowed in the worker

- if we deploy multiple instances in a single runtime, then some application may take more time to process which will affect the performance of other application in the same runtime.

Also it is hard to manage application when they are deployed in the single runtime

Q. single worker Vs multiple worker deployment

- Deploying an application to more than on worker gives the advantage of high availability and zero downtime

Q. What is zero down time ? How does it work?

- **Downtime** in the context of computers refers to that period of time when a computer system is inactive or dominant

- Zero Downtime describes a site without service interruption. Zero Downtime - which means the systems should made available 24x7 365 days. This very critical for Finance, Banking,. Transport and major service utilities that are provided by Government/ Private sectors.

## VPC

* Create a virtual private cloud
* It is used to create an isolated and dedicated network specific to a customer
* To have VPC user must create a separate account and different n/w for a cloud hub (IPSec)
* It creates a dedicated resources to the user
* Provides DLB

Why VPC?

* It is used to create a secure communication between mulesoft and its customer’s server/ data center
* Create a **secure channel** between customer n/w and mulesoft n/w called IPSec when using a public cloud or create a direct connection that is provided by AWS
* **Enables to get a vanity url**: a vanity URL is a long URL that has been converted into a customized short link. A vanity URL can also be known as a branded Link or a custom short URL
* Provides **dedicated load balancer**

**Shared Load Balancer Vs Dedicated Load Balancer**

**SLB**

* SLB url always have xxx.eu.cloudhub.io
* SLB has external IP
* SLB do not allow vanity url
* Do not have custom option for url redirect
* It is shared with multiple instances of application you deploy
* It has a rate limit to make a request. After the limit is exceeded the it throws 503 error

**DLB**

* If you don’t want to expose xxx.eu.cloudhub.io to your customer then use DLB
* Used for url redirect
* Provides vanity url capability
* Provide public url
* Enables two way SSL to store certificate
* Used for IP white listing
* It has internal and external IP

How VPC works

* When you subscribe for a VPC then you will be provided with two VPC environment each with one dedicated load balancer.
* First we need to create a VPC account.
* **Create a secure tunnel** via IPSec or dedicated connetction
* Then configure the firewall
* Then deploy application in VPC zone
* *If the VPC is provided by the AWS then you can add an extra load balancer as well*

**Availability Zones :** Amazon EC2 is hosted in multiple locations world-wide. These locations are composed of Regions, Availability Zones, Local Zones, and Wavelength Zones. Each Region is a separate geographic area. Availability Zones are multiple, isolated locations within each Region. Each region has three data center

Adv of VPC:

* Security
* Vanity url
* Provides the ability to block certain (implementation) URL for security reasons like block customer to access the implementation url and redirect to proxy url
* Maintenance according to the customer choice
* Provide flexibility to the business
* DLB is available

Q why one should choose private cloud over mule cloud hub

## Hybrid

Uses Mulesoft hosted control plane and Customer hosted runtime plane: on prem or cloud

Servers are created. All the servers created must have same Mule runtime engine version and Runtime Manager agent version

Ways to deploy application in a hybrid model deployment

* Standalone server
* Cluster based: The nodes are aware of one another and share common information and synchronize statuses
* Server group: It is a group of server that work in isolation . Instances of the applications are completely isolated from each other

|  |  |
| --- | --- |
| Customer Hosted Runtime Plane | Mulesoft Hosted Runtime Plane |
| * Provides the option to change the memory capacity | * Cannot select worker |
| * Cannot externalize log to anypoint so need to use 3rd party system | * Cannot change region |
|  | * Enables flow monitoring: how many time an API is escuted |
|  |  |

Q How to achieve vertical and horizontal scaling ?

* It is managed by customer
* Allows to increase node
* Customer hosted design

**Hazelcast:**

* It is a caching mechanism for mule runtime.
* It is a distribution memory grid that come along with standalone
* It is a caching solution
* popular choice for MuleSoft users who need to cache data in an external in-memory cluster for increased performance and scale of their MuleSoft deployments. Hazelcast also offers an easy way to pass data or share state from one service to another
* Features that it provides are:
  + Load balancing
  + Replication of application in clusters

Load balancer: It ping all the server in the network every second. If response is received from the servers than it means the servers are active or alive.

Q What is the maximum number of nodes for cluster

* 8 nodes per cluster

Q Why 8 nodes per cluster

Q Why to use domain project

* It is used to share resources among the project
* Advantages: minimum workload, easy to maintain
* Requirement: To use domain project we need to deploy the domain project in every node that is created in a cluster or server group

Why cluster?

* There can be maximum of 8 servers in a cluster
* Synchronize the status between the server. So server know about each other

**Anypoint Runtime Management Agent**

* Responsible to send data to control place

Types of cluster:

* Unicast: to setup unicast we need to provide the IP and all the servers in the clusters should have the same IP. When we need to add a new node then first we have to stop the cluster and then add a new one
* Multicast: to setup, we need not provide the IP it automatically set the same IP for all the nodes in a cluster. Advantage is that we can add nodes dynamically without having to stop the cluster unlike Unicast.

Advantages of domain project:

* Share persistent storage
* Allows to deploy domain project, This provides consistency between application. Example When you need to change the token we can make change in the domain project and it is reflected in all the project.
* Share resources. Example share the port between application

Advantages of hybrid project:

* No core limitation like mulesoft runtime deployment. Here, minimum core size is 0.02
* Can deploy multiple application in same runtime
* This reduces the licensing cost. Allows to share resources

Disadvantage of hybrid

* Auto scaling is not possible. To scale must design a pipeline
* Reduced out of the box monitoring

**Load Balancing**

Two types of load balancing:

* Hard load balancing – done by physical device
* Soft Load Balancing – done via software

Advantage of load balancing :

* Efficiency – if one server is down then another server can complete the request
* Reduced downtime – incase of the server failure customer application processing is not interrupted
* Flexibility – easy to add or remove server from cluster
* Scalable- allows to add or remove the server as requirement

## RTF

Customer hosted runtime plane

Mulesoft hosted control plane

Disadvantage of hybrid

* Application isolation is not available
* Need to manually update versions or patches
* Need an external load balancer
* Does not support mule 3 and mule 4 execution

Why RTF?

* high availability, automatic failover, rolling upgrade, Zero downtime upgrades
* -over comes the disadvantages of the hybrid
* we can deploy multiple versions of mule-runtime

It is used by the organization that have already setup an standalone infrastructure

RTF

* The runtime plane is hosted on the customer on-prem or the provider like AWS, Azure
* based on docker and kubernetes to achieve portability and horizontal scalability
* The architecture of the Runtime Fabric cluster is composed of two types of nodes: controllers and workers.
* Runtime fabric is a collection of VM that work together to form a cluster. Each VM in the cluster serves as either a **“Controller”** node or **“Worker”** node
* Kubernetes controller/ Controller, here the orchestration, distributed database, and internal load-balancing capabilities are done. In addition, most control plane communication with AnyPoint Platform also happens here
* The RTF worker node is where regular Mule applications will get deployed as pods. The more workers you have, the more highly-available and performant your cluster will be

RTF deployment: differ in the number of Controllers / Worker required. This is necessary to achieve fault tolerance and high availability.

* 1. Development:
     + Controller nodes- 1
     + Worker nodes: 2
  2. Production
     + Controller nodes- 3
     + Worker nodes: 3

## PCE

* Private Cloud Edition (PCE) one can run and manage Mule applications on their local servers using their required security policies .
* This is a single-tenant version of Mulesoft which offers full control of the data within the organizations own data centers.
* The control plane appliance can be installed onto the organizations own data centers, such as on-prem or Azure or AWS.
* Thus allows to configure and run Anypoint Platform software and maintain all data storage, processing, and transmission.
* Anypoint Platform PCE uses Docker and Kubernetes to provide built-in high availability and scalability.
* In this the control plane components like design center and management are released into docker containers. Those docker containers are managed by the Kubernetes. The Runtime plane has the bare metal infrastructure. On top of it are the VMs and all the applications are deployed in those VM. The control plane uses the Docker and Kubernetes while the runtime plane is the collection of the clusters.

**Why PCE?**

* If an organization has a strict compliance law on data processing within their own data centers then PCE is suitable. In such a case the organizations won’t even allow the flow of meta data outside the premise.
* Example industries such as banks, insurance corps, healthe care and government organization have scenarios where they want to more tightly manage their data.

**Features that are available**



* **Supported Configurations**
  + 3- Node configuration
    - minimum number of nodes – 3
    - Each node has a application and services and database and file store
    - But one db is used as master and the remaining is used as hot standby instances
    - must configure a load balancer to use round-robin distribution
  + 6-Node configuration
    - 3 nodes are used for application and services
    - Other 3 nodes are used for database and file store
    - Each node has a application and services and database and file store
    - But one one db is used as master and the remaining is used as hot standby instances
    - must configure a load balancer to use round-robin distribution

Additional configuration that need to be done are:

* + Load Balancer
  + Configure Backup and Restore
  + Provide Platform IP or Domain
  + Configure Log Forwarding
  + Set up SMTP
  + SSL Certificate
  + Add a Custom Disclaimer Message
* **Pros**
  + Control Plane and Runtime plane are secured by the organization
  + No flow of metadata to the Mulesoft hosted cloud
  + All the exchange assets are strictly remain on premise
  + High performance and scalability due to use of kubernetes
* **Cons**
  + Requires expertise to manage the control plane
  + Sometimes its difficult to resolve issues

## PCF

Pivotal Cloud Foundry (PCF) uses cloud factory to deploy the runtime engine

App 1 
Build Pack 
App2 Apps 
Cm tainer e rwir• 
, Docker , 
Infrastructure 

* Pivotal cloud foundary can be hosted on customer Data center or AWS or GCP or on any cloud providers
* If we have an infrastructure On top of ‘Infrastructure’ ,PCF provides own  container engine which is  like Garden or Docker or Kubernetes
* On top of container engine ,we  deploy our containerized applications.
* PCF determines mule applications with the help of ‘Build Pack’
* Build Pack’ is to determine what kind of application it is , mulesoft has created its PCF build packs , using those build packs PCF determines those are mule applications and they would require a mule runtime and accordingly it will help to deploy the application

‘

# Security

## Network Security

It is a set of rules and configuration designed to protect the integrity confidentiality and accessibility of computer networks and data using both software technologies.

**How to enhance the network security from the client side**

* Install software on the load balancer to **monitor** the traffic
* You can **capture** the incoming and outgoing request and response or look for specific information
* So customization on the load balancer is possible

Specific customization as per the user need – use VPC

Two system interact with each other before that they authenticate each other. There are different ways to do: one way SSL, two way SSL is done using **handshake**

## Data level Security

Data can be encrypted to provide security. This happens in the system level

* When different Data center needs to interact they authenticate each other via internal certificate
* Example the level of security need for the bank is more than that of social n/w apps like facebook

Adding a security beyond anypoint policies provides following benefits

* More control
* More secure
* Branding

# Runtime Architecture

## Mule 3 Runtime Architecture

* Based on SEDA- Staged Event Driven Architecture
* Decouples flow into three phases
  + Receiving: components the trigger the flow. Eg listener
  + Processing: process the data or payloads
  + Dispatching: outbound components
* It uses three different thread pools for each of the phases mentioned above
* In case of mule 3, developers need to configure the thread pools
* And to manage these thread pools SEDA is executed in back

Advantages of SEDA

* Developers can configure the thread pool as per their need
* Allocate resources for each phase
* Allocate threads for each phase. Recommended: 16 threads per thread pool. But user can have more as well
* User can configure: thread pools, queue, and behavior

Disadvantage

* Developers need to spend time to analyze and configure the threads
* Improper configuration of the threads and resources will affect the over all performance of the system

To over come these disadvantages new version was released by mulesoft

## Mule 4 Runtime Architecture

* Mule 4 **eradicates the need for manual thread pool** configuration as this is done automatically by the Mule runtime.
* IT is based on – reactive programming
* Allows concurrent processing :
  + Suppose if there are 100 request then
  + First 25 request are processed by receiving thread and after it complets its works it is passed to next processing thread i.e. dispatching threads.
  + Now while the dispatch thread process the received request the received thread takes next 25 request and process it
* Why Reactive programming?
  + Customers expectations change frequently
  + Now a days request must be completed in milliseconds due to the excessive use of IoT devices and require 100% uptime
* Features of Reactive programming
  + Responsive: enable rapid and consistent response time
  + Resilient : over come any failover. It is achieved via containment, isolation, replication
  + Elastic: should be able to handle the unexpected traffic
  + Message Driven: Reactive Systems rely on [asynchronous](https://www.reactivemanifesto.org/glossary#Asynchronous) [message-passing](https://www.reactivemanifesto.org/glossary#Message-Driven) to establish a boundary between components that ensures loose coupling, isolation and location transparency
* Reactive Programming facilitates **Non-Blocking execution** of event and processors
* <https://muleforbeginners.wordpress.com/2016/08/05/non-blocking-processing-strategy/#:~:text=The%20difference%20is%20that%20Mule,threads%20are%20available%20when%20needed>.
* Mule 4 design concept uses the best from the Iterator and Observer design patterns, and functional programming.
* The **Iterator pattern** gives the consumer the power and control over when to consume data.
* The **Observer pattern** gives the publisher power over when to push data.
* **Reactive combines** the two to get the beauty of data being
* Uses **NIO selector pool** for non blocking execution
* Back Pressure features:

1. **automatically configured** so that the event source receives a back pressure signal when all threads are currently executing and no free threads remain in a required thread pool

if subscriber is not available – it send 503 error message – this notifies the listener that the server is busy

1. **Manual configured**
   * Allows developer to configure the back pressure signal
   * This is achieved by using the Max Concurrency attributes
   * It affects the number of events that can pass through the event processor per second.

**Processing Type:** type CPU lite, CPU intensive and blocking IO

1. CPU\_LITE
   * Used for the tasks that takes at most 10ms to execute.
   * No blocking IO operations should be executed here
   * Number of threads = (Thread Default size is ) 2 \* cores
   * Logger, set payload
2. CPU\_INTENSIVE
   * For tasks that take more than 10ms to execute (duration is not enforced, but misclassifying tasks has bad consequences).
   * Typically for transformations, encrypt/decrypt, heavy computation, etc. like data weave and scripting
   * Non-blocking IO operations should be executed here.
   * Number of threads = (Thread Default size is ) 2 \* cores
3. BLOCKING\_IO
   * All blocking IO operations should happen here.
   * Significantly larger than the other pools, as most threads here are expected to be in a blocked state.
   * Default size comes from a formula that considers the available memory, the default size of the streaming buffers and other concepts.
   * Database, I/O opertions components like FTP

**Thread pools in mule 4**

In mule 4.1 and 4.2 the threads are based on processing type. All the threads are located in the separate pool

 A running Mule application will pull threads from each of those pools as events pass through its processors. The consequence of this is that a single flow may run in multiple threads.

In mule 4.3 All the threads are located in the centralized pool. **Advantage**  improve the Mule runtime’s auto-tuning feature and make better use of available resources. We can configure pooling strategy in scheduler-conf.properties file

**HTTP thread pools**

The http **selector pool** has two types of threads:

* 1. Grizzly **Shared thread** pool : used for HTTP Listener
  2. Grizzly **Dedicated thread** pool: used for HTTP requester

**Thread switching**

* Based on the processing type thread switching will happen. But when an IO or CPU\_INTENSIVE task is followed by a CPU\_LIGHT task like logger then instead of switching the thread the same thread will execute the CPU lite task. The advantage is to optimize latency of switching between the processes
* Proactor pattern: Proactor is a design pattern for asynchronous execution.
* Scheduler execution service which is responsible for assigning threads to different event processors

**JVM Tuning**

* We can modify Heap Memory and MetaSpace Memory for better performance
* **Heap size** : enables to alter the memory size. Allocate more memory when we have to run large payload or when multiple programs are running in a runtime
* **MetaSpaceSize**:  Allocate more memory for internal processing. Incase to load more class or class loaders.

**Garbage collection** (GC) it keeps track of all the free memory spance so that is ca be used by other processes. in the Java virtual machine (JVM) to automatically determine what memory is no longer being used by a Java application and to recycle this memory for other uses.

**Parallel Garbage collector** (also known as throughput collector) is a garbage collector which uses multiple threads to speed up garbage collection.

**Concurrent Mark and Sweep (CMS)** collector is designed for applications that prefershorter garbage collection pauses and that can afford to share processor resources with the garbage collector while the application is running