MULESOFT ARCHITECTURE:

Deployment Models and Resource Consumption

PRACTIC LEARNING

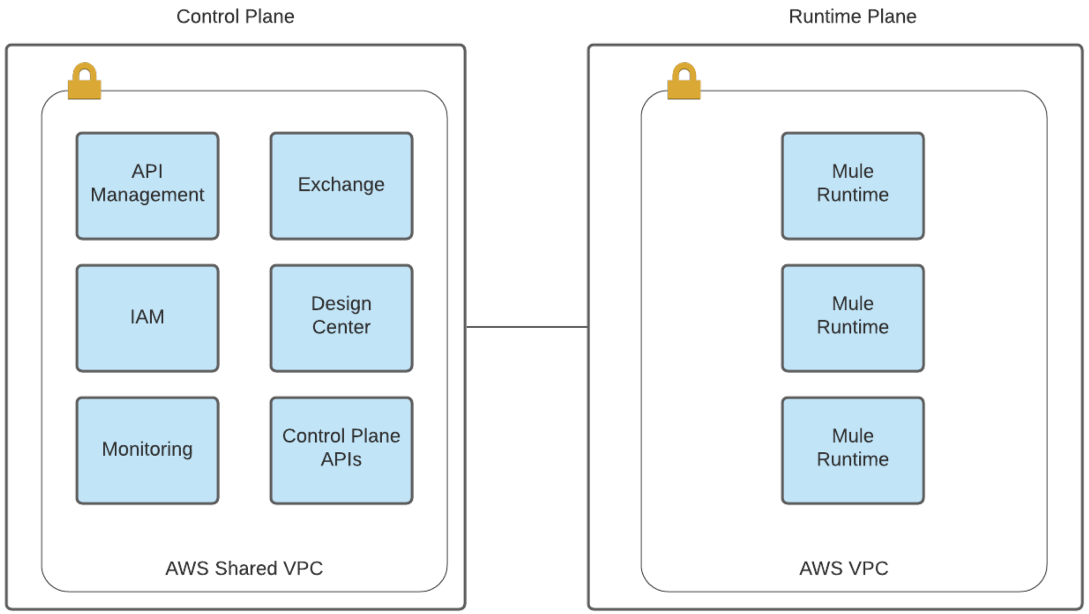
Deployment Models and Resource Consumption Introduction

1. Runtime Plane and Control Plane overview
2. Deployment options
3. Deployment types
4. Deploying to MuleSoft’s cloud
5. Sizing and scaling applications
6. Deploying in a hybrid and on-prem model

Runtime Plane and Control Plane

MuleSoft Core Concept: Runtime Plane vs. Control Plane

- Runtime Plane is where applications/APIs are deployed and run
  - Runs on an AWS VPC if deploying to MuleSoft's Runtime Plane or your network/servers if you deploy in a hybrid or on-prem model
- Control Plane is used to manage the Anypoint Platform and the applications/APIs that are deployed to the Runtime Plane



Control Plane

- Accessed via browser or API
- Contains a well-documented set of APIs on Anypoint Exchange
  - Great for systematic access to Anypoint Access Management, API Manager, deployments, and much more
- Lives in a shared AWS VPC on the backend
- Communicates over the internet

MuleSoft Deployment Options

Determine the Runtime Plane from the available deployment options MuleSoft offers

Deploy to MuleSoft's Cloud

- CloudHub
- GovCloud

Deploy to Managed Servers

- Runtime Fabric
- Hybrid
- Private Cloud Edition (PCE)
- On-prem

MuleSoft Deployment Types

There are two types of deployments MuleSoft offers

Use of the Control Plane

- CloudHub
- GovCloud
- Runtime Fabric
- Hybrid
- Private Cloud Edition (PCE)

Headless (No Control Plane)

- On-prem

MuleSoft Deployment Options

Deployment Target	Description	Leverages MuleSoft's Cloud	Headless
CloudHub	Deploy to MuleSoft's commercial cloud where MuleSoft manages the infrastructure. This is the preferred option for deployments if possible.	Yes	No
GovCloud	Deploy to MuleSoft's government cloud that is FedRAMP certified. Good for government entities who have FedRAMP regulations. Contains a limited subset of features from the MuleSoft Control Plane.	Yes	No
Runtime Fabric	Deploy to your servers with the speed and flexibility of the cloud using a Docker and Kubernetes deployment solution MuleSoft offers.	No	No
Hybrid	Deploy to your servers and manage your servers and applications using MuleSoft's Control Plane.	No	No
PCE	Deploy the MuleSoft runtimes and Control Plane to your servers. Good for customers with tight security or networking requirements. Contains a limited subset of features from the MuleSoft Control Plane.	No	No
On-prem	Deploy to your servers in a headless model without the MuleSoft Control Plane.	No	Yes

MuleSoft Cloud Deployments

CloudHub and GovCloud

- Runtime Plane in MuleSoft’s cloud
- MuleSoft manages the infrastructure
  - Amazon EC2 of various sizes on the backend
- Uses vCores (allocated memory and CPU) to provision Mule worker capacity

Scaling in MuleSoft’s Cloud

- Vertical scaling
  - Increase the vCores allocated to a Mule worker
  - Attempt to vertically scale only when necessary, such as in the event of an OOM error
- Horizontal Scaling
  - Increase the number of load balanced Mule workers (replicas) assigned to a MuleSoft application
  - Makes better use of microservices architecture
  - Most optimized way to use vCores
  - When high availability is needed, use at least 2 workers to create a cluster

MuleSoft Cloud Licensing and Cost Considerations

- CloudHub deployments run on vCores
- MuleSoft license prices based on non-prod and Prod vCores
- As a MuleSoft architect, you are often tasked with most efficiently utilizing vCores
- **Horizontally scaling at 0.1 and 0.2 vCores is the most optimized use of vCores for performance and cost**

Worker Size	Memory Allocation	EC2 Instance
0.1 vCores	500MB	t2.micro
0.2 vCores	1GB	t2.small
1 vCore	1.5GB	m3.medium
2 vCores	3.5GB	m3.large
4 vCores	7.5GB	m3.xlarge
8 vCores	15GB	m3.2xlarge
16 vCores	32GB	m3.4xlarge

Sizing Applications in MuleSoft's Cloud

- **0.1 vCores** – Good for lightweight APIs such as system APIs, little or no processing, proxies, and passthroughs
- **0.2 vCores** – Good for lightweight processing, system APIs, process APIs, experience APIs, transformations, and data enrichment
- **1+ vCores** – Useful for processing large payloads, heavyweight processing needs, and batch processing. Try to avoid 1+ vCores allocated to your applications by adapting architecture or horizontal scaling at lower vCores

Sizing Considerations

- Sizing is completed in 2 phases
  1. An estimation before a project starts based on requirements and/or design
  2. Finalize worker size based on QA testing including load and performance testing
- To estimate sizing before a project begins
  - Use the 6 inputs to sizing shown in the table to the right
  - Notes on using the table
    - The table will help to produce an estimate, not get exact worker size
    - Use a combination of all 6 inputs to sizing and choose the highest value range for your estimation

Input to Sizing	**Options**		
Application Type	🟩 Proxy	🟧 API	🟥 Application
Transactions Per Second	🟩 1	🟧 10	🟥 50+
Payload Size	🟩 100KB	🟧 500KB	🟥 1MB+
Response Time	🟩 High latency	🟧 Medium latency	🟥 Low latency
Complexity	🟩 Low	🟧 Medium	🟥 High
High Availability	🟩 No – 1 worker	🟧 Yes – 2 workers	🟥 Yes – 3+ workers

Legend
🟩 0.1 – 0.4 vCores
🟧 0.2 - 0.6 vCores
🟥 0.4+ vCores

Sizing Scenarios

Application Sizing Scenario #1

An organization wants to build an API to centralize access to Salesforce. The API will have less than 1 request/second sent to it, no transformations, and deal with small payloads. The architect must estimate the number of vCores this API uses in the Dev, QA, and Production environments when deployed to CloudHub.

Capacity Estimations

1. **Dev** - 0.1 vCores x 1 worker = 0.1 vCores
2. **QA** – 0.1 vCores x 1 worker = 0.1 vCores
3. **Production** – 0.1 vCores x 2 workers = 0.2 vCores

Application Sizing Scenario #2

An organization wants to build an API to centralize access to Salesforce. The API should be able to handle 10 requests/second sent to it, no transformations, and deal with medium sized payloads. The architect must estimate the number of vCores this API uses in the Dev, QA, and Production environments when deployed to CloudHub.

Capacity Estimations

1. **Dev** - 0.1 vCores x 1 worker = 0.1 vCores
2. **QA** – 0.2 vCores x 1 worker = 0.2 vCores
3. **Production** – 0.2 vCores x 2 workers = 0.4 vCores

Application Sizing Scenario #3

An organization has a need to deliver employee data to a web application from 2 different data sources and combine them into a single unified response for the web application to consume. The architect has designed 2 simple System APIs that a Process API makes requests to, enriches the data, and transforms the data to be consumed by an Experience API with lightweight transformation. The architect must estimate the number of vCores this API uses in the Dev, QA, and Production environments when deployed to CloudHub.

Environment	System API #1	System API #2	Process API	Experience API	Total
Dev	0.1 vCores x 1 worker = 0.1 vCores	0.1 vCores x 1 worker = 0.1 vCores	0.1 vCores x 1 worker = 0.1 vCores	0.1 vCores x 1 worker = 0.1 vCores	**0.4 vCores**
QA	0.1 vCores x 1 worker = 0.1 vCores	0.1 vCores x 1 worker = 0.1 vCores	0.2 vCores x 1 worker = 0.2 vCores	0.1 vCores x 1 worker = 0.1 vCores	**0.5 vCores**
Production	0.1 vCores x 2 workers = 0.2 vCores	0.1 vCores x 2 workers = 0.2 vCores	0.2 vCores x 2 workers = 0.4 vCores	0.1 vCores x 2 workers = 0.2 vCores	**1 vCores**

Application Sizing Scenario #4

An organization has a need to build an automated, scheduled process that transfers data from a database to SAP. The solution must process and transform hundreds of thousands of records 4 times per day. The architect has designed a batch process as a solution to this business need, and must estimate the number of vCores this API uses in the Dev, QA, and Production environments when deployed to CloudHub.

Capacity Estimations

1. **Dev** - 0.1 vCores x 1 worker = 0.1 vCores
2. **QA** – 1 vCores x 1 worker = 1 vCores
3. **Production** – 1 vCores x 1 workers = 1 vCores

MuleSoft Hybrid and On-Prem Deployments

Deploying MuleSoft to Managed Servers

- Runtime plane becomes servers that an organization manages
- Hybrid model uses MuleSoft Control Plane for management
  - Strongly recommend connecting on-prem runtimes to the MuleSoft Control Plane
- On-prem model is deployed using a headless strategy without the MuleSoft Control Plane
- Uses cores (allocated CPU) to provision Mule worker capacity
  - Similar to CloudHub vCores, but you have more control of allocated memory
    - MuleSoft can help convert between cores and vCores
- When deploying to managed servers, continue to utilize the same principles as deploying to the cloud
  - Use microservices based architecture
  - Use smaller core allocation for APIs and applications
  - Set up infrastructure and architecture to scale horizontally

Servers and Clusters

- Servers
  - Create servers to use the MuleSoft Control Plane to manage and deploy on-prem runtimes
    - Allows for easier deployments, alerting, and some of the governance of the MuleSoft Control Plane
- Clusters
  - Cluster of 2 or more servers
    - For production, use 3+ servers in a cluster
  - Benefits
    - HA
    - Shared memory leading to shared state
    - Load balancing
    - Application awareness across servers

Infrastructure Architecture

- Ensure that servers are large enough to handle multiple Mule applications
- As more applications are deployed to a single server, must scale up the server, or add more servers to the cluster
- Architecture recommendations
  - Cluster for pre-production and production environments
  - Make sure you have HA built in at the infrastructure layer
    - 3+ servers in Production
  - Load balance incoming requests
  - Ensure networking allows for ingress and egress traffic
    - On-prem runtimes need to communicate via internet 2-way SSL with the Control Plane
  - Base infrastructure on sizing and throughput needs

Real World Runtime Plane Scenarios

- CloudHub is the recommended Runtime Plane
  - The most capable, fully featured deployment option
- Deploying to multiple destinations is common
  - Example: deploy to CloudHub and using the hybrid model to managed servers
  - Good reasons to deploy to multiple Runtime Planes include
    - Deploy to CloudHub for public facing APIs, and deploy in a hybrid model with the Runtime Plane in your data center alongside data for quick access to secure data
    - By default, deploy all applications to CloudHub. When necessary, deploy to a Runtime Plane in the data center in a hybrid model for internal facing APIs, Process APIs, and System APIs

Deployment Model and Resource Consumption Summary

- Runtime Plane and Control Plane overview
- Deployment options and types
- Cloud deployment considerations
- Sizing and scaling applications
- Hybrid and on-prem deployment considerations

Additional Reading

- https://docs.mulesoft.com/runtime-manager/cloudhub-architecture
- https://docs.mulesoft.com/mule-runtime/4.3/mule-standalone
- https://docs.mulesoft.com/runtime-manager/servers-about
- https://docs.mulesoft.com/runtime-manager/cluster-about
- https://docs.mulesoft.com/mule-runtime/4.3/mule-high-availability-ha-clusters
- https://docs.mulesoft.com/runtime-manager/monitoring