**Tools Evaluation Approach and List for Fleet Manpower Requirements Determination (FMRD)**

**Frame Work for Selecting Infrastructure Tools:**

The following outlines our approach to select infrastructure tools. It discusses how to find potential tools, determine tool requirements, analyze tool performance, and select tools.

**Understanding Use case for Tools:**

We begin infrastructure tools evaluation by considering what a system needs to do and understanding what functionality a system is delivering to the user. Next, we talk with stakeholders, engineers, and functional teams to understand the types of technologies that are needed.

Potential Questions to Ask:

* What type of computation will your application be using? Does it need to support web applications, databases, heavy ETL processing, machine learning, etc.?
* Will tools be coupled or decoupled so that tools with similar use cases can be easily changed?
* Are the workloads heavily computational, work with big data, transactional work, etc?

**Gathering Potential Tools:**

Our list of potential tools starts by researching industry and company-wide best practices for tools that provide the appropriate functionality required. Next, we meet with engineers and internal subject matter experts on what tools are currently in use and others which are available that meet capability requirements.

**Understanding System Requirements:**

All systems come with certain requirements. Requirement are: security, cost, performance, reliability, redundancy, and maintenance costs. We define what is needed from each of these elements or others relevant to one’s business.

Security – These are the public, federal, community, or internally defined standards that need to be met to provide to the appropriate physical and cyber infrastructure protection.

Cost – These are the costs to procure, operate and maintain the tool (i.e. licenses, service level agreements, or open source).

Performance – These are the tool metrics on overall system speed, data transfer speeds, and data processing rates required to meet capabilities.

Reliability/Redundancy – These are the uptime the system needs, critical nature of the tool, risk tolerance for losing data, restoration time, and infrastructure importance.

Maintenance – These factors include tool licenses, man power resources are needed to maintain it, frequency of upgrades, patches, and infrastructure management.

**Understanding Tools Capabilities:**

After a list of tools is created we evaluate them in several ways. The below list is not comprehensive, but provides a starting point for gathering data on a tools capabilities.

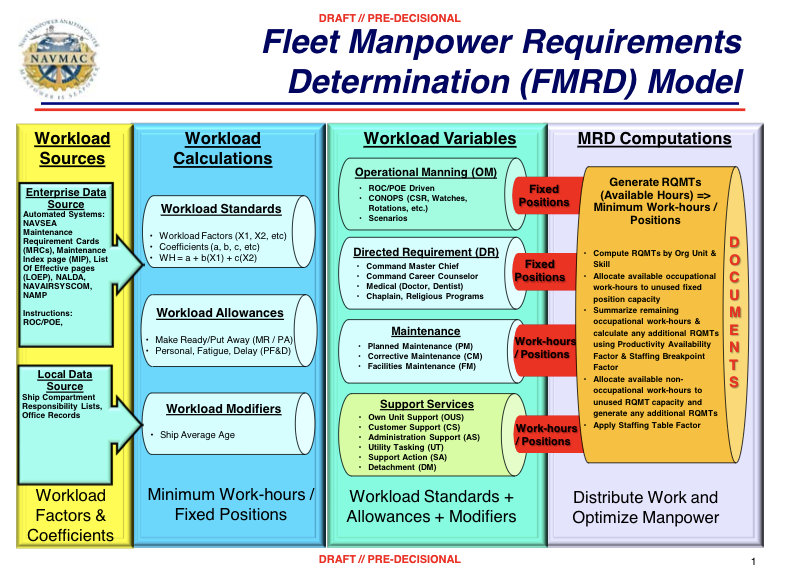
* Reading white papers and use cases describing how a functionality was used and why it was picked.
* Reading the tools documentation to understand exactly what it can and can’t do according to its authors or maintainers.
* Understanding any contract or license that might come with the tool to understand limitations or restrictions on how you may use the tool.
* Using the tool by building rapid prototypes helps develop a sense of the tool capabilities, strengths, and weaknesses.

**Comparison of Tool Capabilities and Requirements**

After relevant data about the tools is gathered an analysis is conducted to determine what capability a tool can provide, the business requirements a tools meets, and the difference between a tools functionality and its requirements. This difference helps determine which tools are a better fit for the overall solution when measured against the other tradeoff metrics discussed above (i.e. performance and cost). Once complete a recommendation is provided on which tool to use or if further analysis is required.

**FMRD**

FMRD program is managed by the Chief of Naval Operations, Total Force Requirements Division and is operated by the Navy Manpower Analysis Center (NAVMAC). NAVMAC is responsible for developing manpower requirements for ships, squadrons, and other deployable units governed by the Required Operational Capability/Projected Operating Environment. The FMRD model is shown in the below graphic.



**Tools Evaluated for FMRD**

**Cloud Based Tools**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tool Type** | **Purpose** | **AWS Tool** | **AWS IL Level** | **Azure Tool** | **Azure IL Level** |
| Container Registry | Docker container registry to store, manage, and deploy Docker container images | Amazon ECR | None | Azure Container Registry | None |
| Relational Database | Store structured data using a traditional relational data model | Amazon RDS  Amazon DynamoDB  Amazon Redshift | IL4 | CosmoDB | IL2 |
| Container Service |  | Amazon Container Service | None | Azure Container Service | IL2 |
| Computational | Runs code when triggered using computer resources needed and lets data from FMRD be executed based on code configuration. | AWS Lambda | None | Azure Functions | None |
| Storage | Data storage provides FMRD with the requirements to handle very large amounts of data and keep scaling to keep up with growth | Amazon S3  Amazon Glacier | IL4  IL4 | Azure Files | None |
| Networking and Content Delivery | Isolate infrastructure built using collection, processing of data, scale request handling capacity and connects network to private virtual network. | Amazon VPC | IL4/IL5 | Azure Virtual Network | IL2 |
| Management | Ability to monitor cloud infrastructure in conjunction with other interdependent components | Amazon CloudWatch  AWS CloudFormation | IL4  IL4 | Azure Monitor | None |
| Analytics | Employment of artificial intelligence to analyze data in large volumes | Amazon Athena  Amazon ES | None  None |  |  |

**Customer Deployed and Managed Tools**

|  |  |
| --- | --- |
| **Tool** | **Purpose** |
| Apache NiFi | Data ingest pipeline |
| Juypter | Analytic notebook allowing data scientists to collaborate on problems. |
| R made for Data Analytics | Statistical modeling code which integrates with other languages, databases and other statistical packages |
| Matlab | Large array of toolboxes and libraries that support various tasks like machine learning and image processing |
| Julia | Open source mathematical language of impressive speed and scalability |
| Caffe | Deep learning library for industry and research, opensource and does not require knowledge of coding |
| Tensorflow | Deep learning tool that supports reinforced learning and other algorithms |
| PyTorch | Python version of Torch, open source code offering dynamic computation graphs, processing variable length inputs and outputs |
| Keras | Deep learning library, providing intuitive API |
| Databricks | Unified analytics platform functioning across the machine learning lifecycle |
| OpenShift | Container registry and management engine |