MATLAB® Online Server

Administration Guide

R2020a, R2020b

How to Contact MathWorks

Latest news:	www.mathworks.com
Sales and services:	www.mathworks.com/sales_and_services
User community:	www.mathworks.com/matlabcentral
Technical support:	www.mathworks.com/support/contact_us
Phone:	508-647-7000

The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

MATLAB Online Server

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See `www.mathworks.com/trademarks` for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

MathWorks products are protected by one or more U.S. patents. Please see `www.mathworks.com/patents` for more information.

Revision History

March 2020	New for Version 1.0 (MATLAB R2020a)
April 2020	Updated (MATLAB R2020a)
July 2020	Updated
September 2020	Updated (MATLAB R2020b)
November 2020	R2020b Update 2

Contents

Part 1: Installation

Part 2: Operations

Part 3: End User Guides

Part 4: Additional Resources

Part 1: Installation

1 Introduction to MATLAB® Online Server

1.1 Overview

1.1.1 Purpose

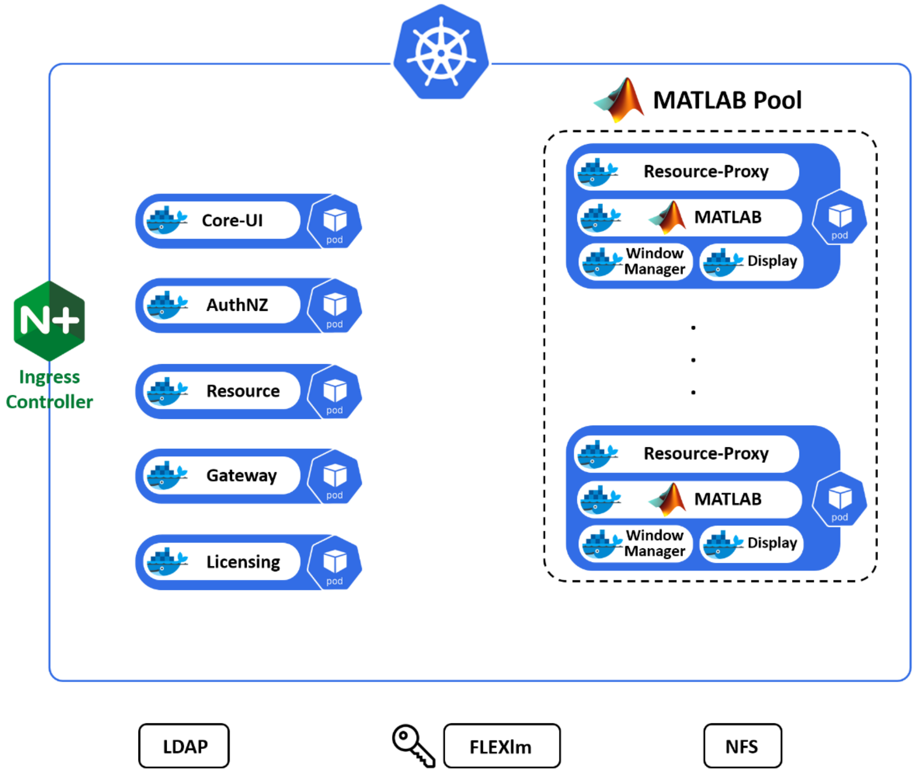
MATLAB® Online Server™ lets you centrally host and manage MATLAB® Online™ for the MATLAB users in your organization, enabling them to access MATLAB from their web browsers. You can integrate MATLAB Online Server with your network file system and authentication protocol (LDAP), configure hardware resources, and define access for your users.

After you configure MATLAB Online Server, your users can access MATLAB Online from a browser by providing their organizational sign-on credentials. MATLAB Online provides an interactive MATLAB desktop environment for iterative analysis and prototyping. This access includes the Live Editor for creating scripts that combine code, output, and formatted text in an executable notebook. MATLAB Online Server can be hosted on your on-premise infrastructure or it can be hosted on public cloud infrastructures but under your private accounts.

See the Glossary for an introduction to many of the terms and third-party applications used when setting up MATLAB Online Server.

1.1.2 Architecture

As shown in the following figure, MATLAB Online Server is deployed as set of containerized applications on Kubernetes, a container orchestration engine for automating the deployment, management, and scaling of containers.



1.1.3 MATLAB Online Server Components

MATLAB Online Server components can be classified into two groups: Core Services and MATLAB Pool.

1.1.3.1 Core Services

The core services consist of:

- Core-UI
  The Core-UI component provides the login screen and other UI capabilities.

- AuthNZ
  The AuthNZ component is mainly responsible for authenticating and authorizing actions of MATLAB Online Server users and some MATLAB Online Server components.

- Resource
  The Resource component enables clients to acquire and release MATLAB instances from the MATLAB Pool. The component maintains information on all MATLAB instances, for example, used or available instances and the users associated with certain instances.

- Gateway
  The Gateway component maps MATLAB Online Server clients to their assigned MATLAB instances. After a mapping is created, the Gateway ensures that any request from a given client is forwarded to its assigned MATLAB instance.

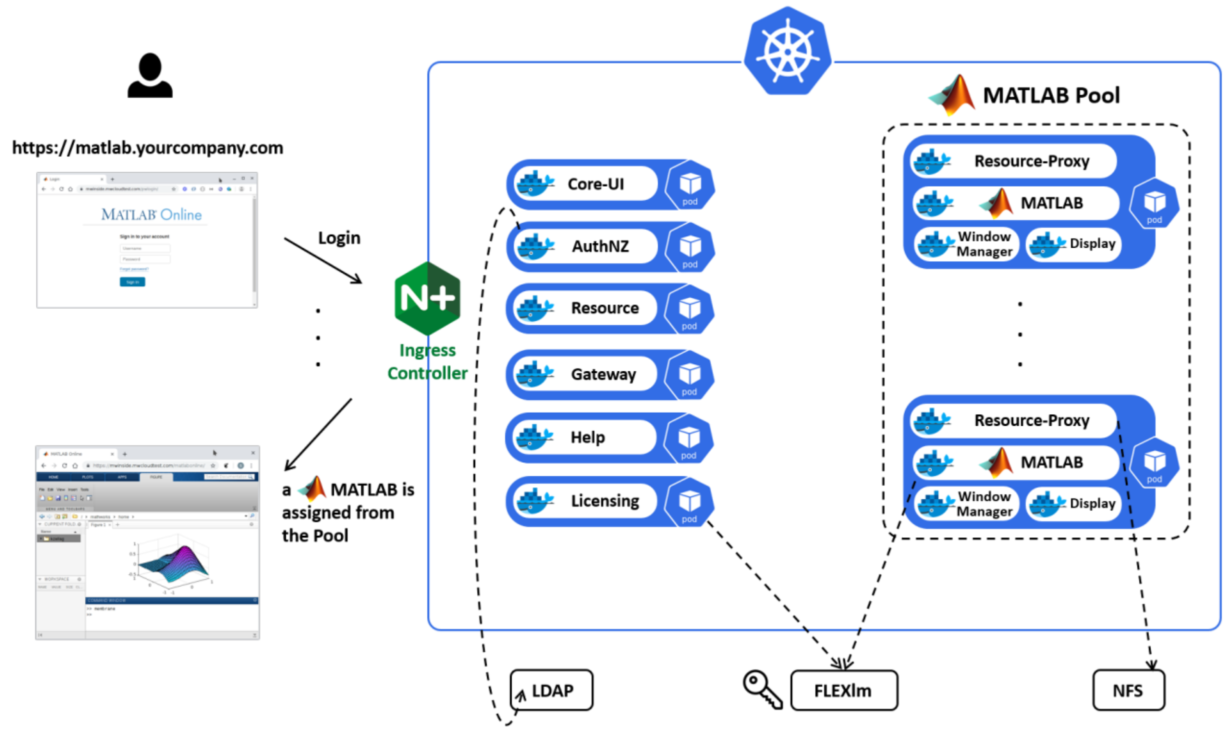
- Licensing
  The Licensing component communicates with the network license manager (FlexNet Publisher, or FLEXlm) server to check for MATLAB Online Server licenses.

1.1.3.2 MATLAB Pool

MATLAB Pool is a collection of pods in which each pod is dedicated to one end user. Each pod is composed of the MATLAB engine and these components: the Resource-Proxy component, which plays the role of a proxy to the MATLAB component, and the Window Manager and Display components, which provide services to the MATLAB component.

1.1.3.3 Workflow

The diagram below illustrates a high-level workflow for when a user signs in to the system.



When a user accesses MATLAB Online from an organization-specific URL, Core-UI serves the landing page containing the request for the end user's user name and password. These credentials are passed to AuthNZ for authentication. If the authorization is successful, the Resource component assigns an available MATLAB instance: a pod with the Resource Proxy, Window Manager, and Display components, and the MATLAB engine.

At this stage, the MATLAB user interface is served to the user and any subsequent requests by the user go through the Gateway component, which forwards them to their assigned MATLAB instance.

All incoming requests to the Kubernetes cluster go through the Nginx Ingress Controller. MATLAB and some MATLAB Online Server components (AuthNZ, Resource, and Gateway) communicate with the Licensing component to check for licenses and the number of concurrent users.

1.1.3.4 Contact MathWorks Support

If you are experiencing technical issues or have difficulty with the instructions when installing or configuring MATLAB Online Server, email matlab-online-server-support@mathworks.com.

Run the following script to collect information that can assist with resolution:

```
sudo ./mosadm gather-system-info
```

This script gathers the system information (Unix system info, present working directory (pwd), disk usage) and outputs it to the terminal.

In addition, have ready as much of the following information as you can:

- MATLAB Online Server License Number
- MATLAB Online Server version
- Debug logs

- MATLAB version and all supplemental toolbox names

1.2 Requirements

1.2.1 Intended Audience

To install and configure MATLAB Online Server, the administrator should have in-depth knowledge and experience installing and managing Linux® systems, configuring LDAP, and configuring networking settings including IP addresses and DNS entries, as well as experience with observing operational systems using logs and monitoring systems.

1.2.2 Supported Operating Systems

MATLAB Online Server is compatible with Linux® only.

Supported operating systems:

- Ubuntu 18.04 LTS
- Ubuntu 16.04 LTS
- Ubuntu 20.04
- RedHat 7.7, 7.8
- CentOS 7.7, 7.8

1.2.3 Kubernetes Support

MATLAB Online Server Version	Kubernetes Support
R2020a and all updates	Kubernetes 1.15
R2020b	Kubernetes 1.15
R2020b Update 2	Kubernetes 1.15, 1.16, 1.17

1.2.4 System Requirements

Each computer on which you install MATLAB Online Server should meet or exceed the minimum hardware recommendations. Note that system requirements for MATLAB Online Server installation can vary based on many factors, including the number of users and the number and size of MATLAB Docker images. If the setup program determines that your computer does not meet the following recommendations, you get a warning, but you can continue with the setup process.

	Processor	CPU/GPU	RAM	Free Disk Space
Single installation	64-bit	8-core, 2.0 GHz or higher	32 GB	50 GB
Per concurrent MATLAB session	64-bit	4-core, 2.0 GHz or higher	8 GB	6 GB

Core count is based on cores. Physical cores can represent actual server hardware or cores on a virtual machine (VM). Hyperthreading is ignored for the purposes of counting cores.

> Note:
>
> You need a minimum of two cores for MATLAB Online Server and Kubernetes. If your computer does not meet these requirements, then the setup program will not install MATLAB Online Server.
>
> In addition, do not install MATLAB Online Server in a cluster where other resource-intensive applications are also running.

1.2.5 License Requirements

1.2.5.1 MATLAB Online Server

MATLAB Online Server is provisioned as a Concurrent license in units of server instances. A server instance corresponds to a URL endpoint within your organization that your users can navigate to for the purpose of accessing MATLAB Online. There are no licensing limits to the number of users you can serve with one server instance, but as noted before, your ability to scale depends on the underlying hardware configuration you select.

MATLAB Online Server is configured to work with the network license manager. You can use the network license manager that you install using the MathWorks product installer or reuse your existing license manager.

1.2.5.2 MATLAB Online

In addition to a MATLAB Online Server license, the use of MATLAB licenses that use a network license manager is required. Examples include Concurrent and Network Named User licenses.

2 Online Install on Single Machine

2.1 Overview

This topic explains the steps you must perform to set up MATLAB Online Server on a single machine for testing and evaluation. To add nodes to the cluster, see Production environment at kubernetes.io.

The MATLAB Online Server single-machine installer assumes that the operating system is one of the Supported Platforms.

After MATLAB Online Server is installed, individual MATLAB instances are hosted by a Docker container, and the MATLAB Online Server stack is managed by Kubernetes.

See the Glossary for an introduction to the terms and processes used when installing MATLAB Online Server.

2.2 Prerequisites

Complete the following software checks before downloading and configuring the MATLAB Online Server installer. The tasks can be done in any order.

- Make sure you are able to run commands using sudo.
- Verify that your environment meets the following system requirements:
    - Is running one of the Supported Platforms.
    - Has Docker 18.09 (with respect to Ubuntu version). For example, apt-get install docker.io (if needed, docker.io = 18.09.7)

2.3 Installation Procedure

2.3.1 Step 1. Install Network License Manager

Install the network license manager for MATLAB on your license server. Configure the server to be accessed over the network.

- You can use an existing network license server or you can create a new installation. If you are performing a new license server installation, follow the instructions in Install Network License Manager with an Internet Connection on the MathWorks website. If your machine is not connected to the Internet, see Install Network License Manager Offline.
- .
- The server and the MATLAB workers need access to their MathWorks licenses. Those licenses can either be co-located on the same license server or on independent servers. For more information, see License Manager Installation and Management on the MathWorks website.
- The MATLAB and server license host name and port(s) can be specified using the MLM_LICENSE_FILE environment variable format of <port>@<host>[,<port>@<other-host>]. The host and port must be accessible from inside the Kubernetes cluster.
- If the license server is configured correctly, then the license service running in the Kubernetes cluster runs without any errors and checks out the license when it starts. If the license is

invalid or the license server cannot be reached, then the license service exits and Kubernetes automatically attempts to restart the license service periodically until it succeeds.

- The other services depend on the license service before they become usable, but they stay running if the license is temporarily unavailable. After an approximately two-hour grace period, the services go into a hibernate state and reject requests with a failed status. The logs of the services indicate when they enter or exit this hibernate or grace period state.

2.3.2 Step 2. Download and Install MATLAB

Follow the Linux instructions in Install Products Online on the MathWorks website to install MATLAB R2020a (or later) on the server. If your machine is not connected to the Internet, there are instructions for installing MATLAB on an offline machine on the MathWorks website. Make sure you install MATLAB into the folder /MATLAB.

2.3.3 Step 3. Download MATLAB Online Server Installer

Follow these steps to download and extract the MATLAB Online Server installer and files.

1. Download the installer
    a. Go to the downloads page: https://www.mathworks.com/downloads
    b. Click on **Download R*XXYY***, where XXYY is the current release, for example, R2020a.
    c. Go to Additional RXXXX Product Downloads and click Get MATLAB Online Server.
    d. Next, click **Download** to download the ZIP file.
2. In the download folder or folder of your choice, extract the installation files. For example:

```
unzip R2020a_matlab_online_server.zip -d ~/
```

Files included in matlab_online_server.zip:

- matlab_online_server/
- data/
- attributions.txt
- install.config
- mosadm
  The mosadm command is the main utility to perform the installation steps. For details on the various operations it can perform, run the following command:

  ```
  ./mosadm help
  ```

- thirdpartylicenses.txt

2.3.4 **Step 4. Install Kubernetes**

2.3.4.1 Download and Install Kubernetes Components

1. Navigate to the extracted folder (~/matlab_online_server) and execute the following command:

   ```
   sudo ./mosadm bootstrap-node
   ```

   This command downloads and installs dependencies and the Kubernetes components. It then initializes a single-node Kubernetes cluster on the machine and customizes the cluster for MATLAB Online Server. It can take several minutes for the installation to finish.

   By default, the docker daemon is configured using the properties in the file `/etc/docker/daemon.json`, and the bootstrap-node command overwrites any customizations.

   See Bootstrap or Reset Single-Node Kubernetes Cluster for more details on the bootstrap process.

   To preserve existing customizations, use the following command instead:

   ```
   sudo ./mosadm bootstrap-node --preserve-docker-config
   ```

2. At this step, you are asked to accept the terms and conditions for installing MATLAB Online Server. If you agree, answer "Y." If you do not, answer "N," and the installer exits. If you do not agree, you cannot install MATLAB Online Server.

3. To verify successful installation, use the following command:

   ```
   sudo ls /etc/kubernetes/admin.conf
   ```

   This file exists after the cluster is created successfully.

4. Check that the pods are in the ready state (1/1 containers running). If you do not see the correct list of pods, check the version number for the install files. Note that it can take several minutes for all pods to come online.

   ```
   sudo kubectl --kubeconfig /etc/kubernetes/admin.conf get pods --all-namespaces
   ```

   The command returns a display similar to the following:

```
ubuntu@ubuntu:~/matlab_online_server$ kubectl get pods --all-namespaces

NAMESPACE      NAME                               READY    STATUS     RESTARTS    AGE

kube-system    coredns-5c98db65d4-9rskh           1/1      Running    2           1m

kube-system    etcd-ubuntu                        1/1      Running    2           1m

kube-system    kube-apiserver-ubuntu              1/1      Running    2           1m

kube-system    kube-controller-manager-ubuntu     1/1      Running    2           1m

kube-system    kube-router-kcbrm                  1/1      Running    2           1m
```

```
kube-system   kube-scheduler-ubuntu             1/1     Running   2          1m
```

If you want your administrator account to be able to access the Kubernetes cluster with kubectl, execute the following to merge the kube-config file into your home folder:

```
sudo ./mosadm merge-kube-config
```

2.3.5 Step 5. Configure Server

Edit the install.config file, located in the folder where you unzipped the installer. Consider creating a backup of install.config before editing to maintain the default settings. In the file, modify the fields described in the following tables.

2.3.5.1 Virtual Machine Configuration

Field Name	Description	Example
DOMAIN_BASE	Domain name used to access the services. Provide the full domain name for this parameter, for example, matlab.mycompany.com. The domain you specify must be registered.	DOMAIN_BASE=matlab.mycompany.com

2.3.5.2 Authentication Configuration

Field Name	Description	Example
ML_PASSWORD	MATLAB Online sign-in password. The user name for this bootstrapping mode is `admin`. **Note:** LDAP configuration is covered in another table.	ML_PASSWORD=password

2.3.5.3 License Configuration

Field Name	Description	Example
`LICENSE_SERVER`	License server details	If you do not already have the network license manager installed on a license server, install it now and return to this step when you are done. Provide the port and the IP address, for example: `LICENSE_SERVER=27000@172.0.0.1`

2.3.5.4 Security Configuration

Consider configuring your environment with Transport Layer Security (TLS). Leverage your organization's best practices for key management, certificate authority, and key rotation for MATLAB Online Server. For a local development environment and during configuration of your environment, it is acceptable to temporarily disable TLS to aid in initial deployment.

Field Name	Description	Example
`IS_TLS_ENABLED`	Flag to enable or disable Transport Layer Security (TLS)	`IS_TLS_ENABLED=false`
`TLS_KEY_FILE`	Path to the TLS key file	If TLS is enabled, provide the path to the TLS key file: `TLS_KEY_FILE=/opt/tls.key`
`TLS_CERT_FILE`	Path to the TLS certification file	If TLS is enabled, provide the path to the TLS certification file: `TLS_CERT_FILE=/opt/tls.cert`

2.3.6 Step 6. Prepare Docker Images

Load all Docker images and files shipped through the installer onto the computer.

1. Load Docker images:

   ```
   sudo ./mosadm load-docker-images
   ```

   Note: If your end user has access to Docker, then sudo is not required.

2. Whether your cluster is new or existing, you must also build, from a local installation, an image containing MATLAB using the following command:

```
sudo ./mosadm build-matlab-image /MATLAB
```

3. To check which images were loaded and built, list all MATLAB Online Server images with the following command:

   ```
   sudo ./mosadm list-docker-images
   ```

2.3.7 Step 7. Install NGINX Ingress Controller

If you are using your own ingress controller, you may skip this step.

Otherwise, install the NGINX ingress controller using the following command:

```
./mosadm install-ingress
```

To verify that the NGINX ingress controller is installed correctly, enter the following command:

```
kubectl get pods --namespace your-namespace
```

Output:

```
NAME                                        READY   STATUS    RESTARTS   AGE
mathworks-nginx-ingress-controller-65cd5f78f5-j8ggd       1/1     Running   0
17m
mathworks-nginx-ingress-default-backend-98c86bdb5-j2g62   1/1     Running   0
17m
```

2.3.8 Step 8. Prepare Helm Charts and Configure Overrides

When the cluster and Docker images are ready to be used, the Helm charts and configuration overrides can be created before deploying the services. Helm is an open source project that helps deploy services into Kubernetes.

1. Copy the necessary Helm charts:

   ```
   ./mosadm copy-helm-charts
   ```

   This command creates a folder named charts in the current folder and copies the charts there.

2. Create the chart configuration files for the deployments:

   ```
   sudo ./mosadm generate-overrides
   ```

   This command creates a folder structure similar to the following:

   ```
   ./overrides/<cluster>/<namespace>/<deployment>.yaml
   ```

2.3.9 Step 9. Deploy MATLAB Online Server

1. After you finish customizing the overrides, execute the following command to install all the MATLAB Online Server components:

   ```
   ./mosadm deploy
   ```

   This command uses Helm to install all the charts, customized with any overrides that you generated in the previous step.

2. Check that the services are all running with kubectl, replacing *your-namespace* with the namespace you used for MATLAB Online Server:

   ```
   sudo kubectl get pods --namespace your-namespace
   ```

The pods running MATLAB can take several minutes to fully start (4/4 containers). It is important to wait until the pods are in the ready state. If any of the containers do not start running, check the license server and the install.config settings. See MATLAB Pod Issues.

```
ubuntu@ubuntu:~/matlab_online_server$ kubectl get pods --namespace your-namespace

Output:

NAME                                                   READY    STATUS     RESTARTS    AGE

your-namespace-core-ui-core-ui-5bd9b7c88b-fqbl4        1/1      Running    0           111s

your-namespace-authnz-authnz-7c7999589c-x74d9          1/1      Running    0           111s

your-namespace-gateway-669dd7865f-t6l74                1/1      Running    0           110s

your-namespace-license-548f7b6655-8skdt                1/1      Running    0           108s

your-namespace-matlab-pool-8c679f76-c4h4n              4/4      Running    0           107s

your-namespace-matlab-pool-8c679f76-cckkp              4/4      Running    0           107s

your-namespace-matlab-pool-ui-579fc8ff49-2bz25         1/1      Running    0           107s

your-namespace-nginx-ingress-controller-kcwwd          1/1      Running    0           104s

your-namespace-nginx-ingress-default-backend-f78c5f7b8-dvxm8

                                                     1/1      Running    0           104s

your-namespace-resource-resource-894447df7-r9ngb       1/1      Running    0           103s
```

2.3.10 Step 10. Verify MATLAB Online Server Installation

1. Make sure the address in the DOMAIN_BASE parameter defined in install.config is registered and can be routed to the IP address of the host machine.

   ```
   ping -c 1 matlab.domain.com
   ```

   This command returns the IP address of the host machine. In some types of installation, the ping does not receive packets. If you have such an installation, this command returns the IP address of the server you ping.

2. Open a browser on another machine and go to the address where you are hosting MATLAB Online:

   matlab.domain.com/matlabonline

   The address is the `DOMAIN_BASE` parameter defined in `install.config`.

3. Your browser window shows a prompt to select a password only. For a single-machine installation, use the user name `admin` and the password `password`.

   Normally, password is the value set in the `ML_PASSWORD` parameter (defined in `install.config`).

If you can now access your internal version of MATLAB Online, installation is complete. If you cannot access MATLAB Online, or you run into any other technical error, contact MathWorks Support.

2.3.11 Next Steps

If you want to apply additional configuration parameters, see Post-Installation Additional Configuration.

3 Offline Install on Single Machine

3.1 Offline Installation Overview

This topic explains the steps you must perform to set up MATLAB Online Server on a single machine in a restricted or air-gapped environment.

MATLAB Online Server installer provides a way to bootstrap a single-node Kubernetes cluster on a node. However, this method pulls many third-party applications, for example, libraries, container images, and configuration files. This workflow might not work when network security measures meant to physically isolate the server from unsecure networks are employed.

Note the following:

- The MATLAB Online Server single-machine installer assumes that the operating system is one of the Supported Platforms.
- After MATLAB Online Server is installed, individual MATLAB instances are hosted by a Docker container, and the MATLAB Online Server stack is managed by Kubernetes.
- See the Glossary for an introduction to the terms and processes used when installing MATLAB Online Server.

3.2 Offline Installation Prerequisites

Complete the following software checks before downloading and configuring the MATLAB Online Server installer. The tasks can be done in any order.

- Make sure you can run commands using sudo.
- Verify that your environment meets the following system requirements:
  - The environment is running one of the Supported Platforms.
  - The environment has Docker 18.09 (with respect to the Ubuntu version). For example, check the version by using apt-get install docker.io (if needed, docker.io = 18.09.7).
- Make sure that the RHEL/Centos node has access to the internet for building the offline installer has the same configuration or is a replica of the offline machine.
- Verify that the RHEL/Centos node is running in air-gapped environment

3.3 Offline Installation Procedure

3.3.1 Step 1. Install Network License Manager

Follow the Linux instructions for installing the network license manager on an offline machine. Configure the server to be accessed over the network.

- You can use an existing network license server or you can create a new installation.

- The server and the MATLAB workers need access to their MathWorks licenses. Those licenses can be either co-located on the same license server or on independent servers. For more information, see License Manager Installation and Management on the MathWorks website.
- The MATLAB and server license host name and port(s) can be specified using the MLM_LICENSE_FILE environment variable format of <port>@<host>[,<port>@<other-host>]. The host and port must be accessible from inside the Kubernetes cluster.
- If the license server is configured correctly, then the license service running in the Kubernetes cluster runs without any errors and checks out the license when it starts. If the license is invalid or the license server cannot be reached, then the license service exits and Kubernetes automatically attempts to restart the license service periodically until it succeeds.
- The other services require that the license server is running, but they will continue to run if the license is temporarily unavailable. After an approximately two-hour grace period, the services go into a hibernation state and reject requests with a failed status. The logs of the services indicate when they enter or exit this hibernation or grace period state.

3.3.2 Step 2. Download MATLAB for Offline Installation

Download MATLAB for offline installation and install on the offline server.

Follow the Linux instructions for installing MATLAB on an offline machine. Make sure you install MATLAB into the folder /MATLAB.

3.3.3 Step 3. Download MATLAB Online Server Installer Files

1. Download MATLAB Online Server to the machine with an internet connection.
   a. Go to the downloads page: https://www.mathworks.com/downloads.
   b. Click on **Download R*XXXXy***, where XXXX is the current release year and y is the current version, either a or b, for example, R2020b.
   c. Go to Additional **R*XXXXy*** Product Downloads and click Get MATLAB Online Server.
   d. Next, click **Download** to download the ZIP file.

2. Unzip the file. For example:

```
unzip R2020b_matlab_online_server.zip -d ~/
```

3.3.4 Step 4. Build Offline Installer

Next, on the same machine, build the installer for MATLAB Online Server with the following command:

```
sudo ./mosadm build-offline-installer
```

This command downloads all the required artifacts for building the installer:

- Kubernetes repo (for Kubernetes executables)

- Yum repos configured on your system (RHEL repos)
- Docker registry
  - Docker hub
  - gcr.io
- Helm charts
  - Github

Refer to Installed Utilities for more details on these artifacts.
After the build command completes, check that the following folders are present:

- <matlab_online_server>/data/kubepacker-rhel-direct-install-1.1.3/files/opt/mw/offlineinstaller
- <matlab_online_server>/data/com.3p.*

If the build was successful, you see the message "Successfully built offline installer." If for some reason the build was not successful, try rebuilding, and if that fails, contact Support.

3.3.5 Step 5. Install MATLAB Online Server on Air-Gapped Machine

1. Copy the <matlab_online_server> folder to the offline server where you want to host MATLAB Online Server. For example:

   ```
   scp -r matlab_online_server
   username@server.com/home/rheluser/matlab_online_server

   scp -r matlab/R2020b username@server.com/MATLAB
   ```

2. Update install.config.
   Open the install.config file and set the OFFLINE_MODE property to true.

3. Install MATLAB Online Server.
   Follow the MATLAB Online Server install workflow starting with Step 4. Install Kubernetes.

3.3.6 Next Steps

This completes the offline installation. Next, continue to Post-Installation Additional Configuration.

4 Install on Cloud-Managed Kubernetes

4.1 Requirements for Installing MATLAB Online Server on Cloud-Managed Kubernetes

These instructions are for installing MATLAB Online Server on cloud-managed Kubernetes.

Make sure your system meets the following requirements:

- Push and pull access to the remote Docker repository
- Existing Kubernetes cluster on Azure AKS, Amazon EKS, or Google GKE
    - Supported node
      Refer to Supported platforms.

    - Support for network policy
      MATLAB Online Server creates NetworkPolicy resources, which are silently ignored if the cluster is not created with network policy enabled. Network policy specifies which pods can communicate with each other. Because it is important from a security standpoint, network policy is enabled by default in MATLAB Online Server.

      If your cluster does not have a container network interface (CNI) that supports network policies, then MATLAB Online Server functions properly but not securely. This configuration is not recommended.

      Refer to documentation from your specific cloud provider for more details on how to enable network policy in the Kubernetes cluster.

4.2 Prerequisite Software Checks

On the machine that will communicate with the managed cluster to deploy MATLAB Online Server (client machine), complete the following software checks before downloading and configuring MATLAB Online Server. These tasks can be done in any order.

- Make sure you can access the cluster’s kubeconfig file from the client machine.
- Make sure the client machine OS is supported. Refer to Supported platforms.
- Install Helm v3.
- Install Kubectl.
- Install Docker 18.09.

4.3 Step 1. Install Network License Manager

Install the network license manager for MATLAB on your license server. Configure the server to be accessed over the network.

- You can use an existing network license server or you can create a new installation. If you are performing a new license server installation, follow the instructions in Install Network License

Manager with an Internet Connection on the MathWorks website. If your machine is not connected to the Internet, see Install Network License Manager Offline.

- The server and the MATLAB workers need access to their MathWorks licenses. Those licenses can be either co-located on the same license server or on independent servers. For more information, see License Manager Installation and Management on the MathWorks website.
- The MATLAB and server license host name and port(s) can be specified using the MLM_LICENSE_FILE environment variable format of <port>@<host>[,<port>@<other-host>]. The host and port must be accessible from inside the Kubernetes cluster.
- If the license server is configured correctly, then the license service running in the Kubernetes cluster runs without any errors and checks out the license when it starts. If the license is invalid or the license server cannot be reached, then the license service exits and Kubernetes automatically attempts to restart the license service periodically until it succeeds.
- The other services require that the license server is running, but they will continue to run if the license is temporarily unavailable. After an approximately two-hour grace period, the services go into a hibernation state and reject requests with a failed status. The logs of the services indicate when they enter or exit this hibernation or grace period state.

4.4 Step 2. Download and Install MATLAB

Follow the Linux instructions in Install Products Online on the MathWorks website to install MATLAB R2020a (or later) on the server. If your machine is not connected to the Internet, there are instructions for installing MATLAB on an offline machine on the MathWorks website. Make sure you install MATLAB into the folder /MATLAB.

4.5 Step 3. Download MATLAB Online Server Installer

Follow the instructions in Download MATLAB Online Server Installer in the Online Install on Single Machine instructions, and then return here and continue with these instructions.

4.6 Step 4. Configure Server

Edit the file `install.config`, located in the folder where you unzipped the installer. Consider creating a backup of `install.config` before editing to keep a copy of the default settings. In the file, modify the fields in the following table.

Field Name	Description	Example
Virtual Machine Configuration		
`DOMAIN_BASE`	Domain name used to access the services.	`DOMAIN_BASE=matlab.mycompany.com`

	Provide the full domain name for this parameter, for example, matlab.mycompany.com. The domain you specify must be registered.	

Authentication Configuration		
ML_PASSWORD	MATLAB Online sign-in password. The user name for this bootstrapping mode is *admin*. NOTE: LDAP configuration is covered below.	ML_PASSWORD=password

License Configuration		
LICENSE_SERVER	License server details	You must have FLEXlm installed. Provide the port and the IP address of the license server, for example: LICENSE_SERVER=27000@172.0.0.1

If you have a remote registry within the company, configure the following information about the registry.

Remote Docker Registry Configuration		
DOCKER_REGISTRY	Remote Docker registry to push Docker images to	Registry name, for example: DOCKER_REGISTRY=myregistry.mycompany.com
DOCKER_REPOSITORY	Remote Docker repository	Repository name, for example: DOCKER_REPOSITORY=mos
IMAGE_PULL_SECRET	User-friendly name for the Kubernetes secret object, which is created from registry credentials (when you deploy MATLAB Online Server). This secret	Secret object name, for example: IMAGE_PULL_SECRET=mymossecret

	allows the nodes to pull images from the remote registry.	

4.7 Step 5. Configure Security

Configure your environment with Transport Layer Security (TLS). See Security Configuration and apply the settings that are appropriate for your organization.

4.8 Step 6. Prepare and Push Docker Images

To allow cluster nodes to use Docker images, you must provide credentials to access the registry. If you have logged in to the remote registry from your client machine at least one time, then the file at ~/.docker/config.json has the registry credentials.

1. Copy the file with the following command:

   ```
   cp ~/.docker/config.json
   ./matlab_online_server/dockerconfig.json
   ```

   Contents from this file are used to create a Kubernetes secret by the name <IMAGE_PULL_SECRET> that is configured in Step 3.

2. Push Docker images to the remote registry (configured in Step 3) with the following command:

   ```
   ./mosadm push-docker-images
   ```

4.9 Step 7. Install NGINX Ingress Controller

If you are using your own ingress controller, you can skip this step.

Otherwise, install the NGINX ingress controller using the following command:

```
./mosadm install-ingress
```

To verify that the NGINX ingress controller is installed correctly, enter the following command:

```
kubectl get pods --namespace your-namespace
```

Output:

```
NAME                                          READY   STATUS    RESTARTS   AGE
mathworks-nginx-ingress-controller-65cd5f78f5-j8ggd       1/1     Running   0
17m
```

```
mathworks-nginx-ingress-default-backend-98c86bdb5-j2g62   1/1     Running   0
17m
```

4.10 Step 8. Prepare Helm Charts and Configure Overrides

When the cluster and Docker images are ready to be used, the Helm charts and configuration overrides can be created before deploying the services. Helm is an open source project that helps deploy services into Kubernetes.

1. Issue the following command to copy the necessary Helm charts:
   ```
   ./mosadm copy-helm-charts
   ```

2. Enter the generate-overrides command with the flag --skip-matlab-image, as shown:

   ```
   ./mosadm generate-overrides --skip-matlab-image
   ```

The mosadm command looks for the MATLAB Docker image locally even when the image is available in the remote Docker repository. So that mosadm can proceed without its having to download the image locally, passing the flag --skip-matlab-image is necessary.

4.11 Step 9. Deploy MATLAB Online Server

1. After you finish customizing the overrides, execute the following command to install all the MATLAB Online Server components:

   ```
   ./mosadm deploy
   ```

   This command uses Helm to install all the charts, customized with any overrides that you generated in the previous step.

2. Check that the services are all running with kubectl, replacing *your-namespace* with the namespace you used for MATLAB Online Server:

   ```
   sudo kubectl get pods --namespace your-namespace
   ```

The pods running MATLAB can take several minutes to fully start (4/4 containers). It is important to wait until the pods are in the ready state. If any of the containers do not start running, check the license server and the install.config settings. See MATLAB Pod Issues.

```
ubuntu@ubuntu:~/matlab_online_server$ kubectl get pods --namespace your-namespace

Output:

NAME                                              READY    STATUS     RESTARTS    AGE

your-namespace-core-ui-core-ui-5bd9b7c88b-fqbl4    1/1      Running    0           111s
```

```
your-namespace-authnz-authnz-7c7999589c-x74d9         1/1     Running   0         111s
your-namespace-gateway-669dd7865f-t6l74               1/1     Running   0         110s
your-namespace-license-548f7b6655-8skdt               1/1     Running   0         108s
your-namespace-matlab-pool-8c679f76-c4h4n             4/4     Running   0         107s
your-namespace-matlab-pool-8c679f76-cckkp             4/4     Running   0         107s
your-namespace-matlab-pool-ui-579fc8ff49-2bz25        1/1     Running   0         107s
your-namespace-nginx-ingress-controller-kcwwd         1/1     Running   0         104s
your-namespace-nginx-ingress-default-backend-f78c5f7b8-dvxm8
                                                   1/1     Running   0          104s
your-namespace-resource-resource-894447df7-r9ngb      1/1     Running   0         103s
```

4.12 Step 10. Connect to MATLAB Online Server

After you have installed MATLAB Online Server successfully, provide a way to access MATLAB Online from the browser.

Run the following command to get the external IP of the cluster, which is required to configure the DNS:

```
kubectl get service --namespace mathworks
```

You can route the `DOMAIN_BASE` parameter, defined in install.config, to the external IP shown in the output of the command above.

4.13 Step 11. Verify MATLAB Online Server Installation

1. Make sure the address in the DOMAIN_BASE parameter defined in install.config is registered and can be routed to the IP address of the host machine.

   ```
   ping -c 1 matlab.domain.com
   ```

   This command returns the IP address of the host machine. In some types of installation, the ping does not receive packets. If you have such an installation, this command returns the IP address of the server you ping.

2. Open a browser on another machine and go to the address where you are hosting MATLAB Online:

   matlab.domain.com/matlabonline

   The address is the `DOMAIN_BASE` parameter defined in `install.config`.

3. Your browser window shows a prompt to select a password only. For a single-machine installation, use the user name `admin` and the password `password`.

   Normally, the password is the value set in the `ML_PASSWORD` parameter (defined in `install.config`).

If you can now access your internal version of MATLAB Online, installation is complete. If you cannot access MATLAB Online, or you run into any other technical error, contact MathWorks Support.

4.14 Next Steps

If you want to apply additional configuration parameters, see Post-Installation Additional Configuration.

5 Install on RedHat OpenShift

5.1 Requirements for Installing MATLAB Online Server on RedHat OpenShift

These instructions are for installing MATLAB Online Server on the Red Hat OpenShift platform.

Make sure your system meets the following requirements:

- Push and pull access to the remote Docker repository
- Existing OpenShift 4.x cluster with:
    - Support for network policy
    - Service account that has access to ”privileged” security context constraints. You can create this access with the following commands (the commands require the oc command tool):

```
>oc create namespace mathworks

>oc create sa custom-sa --namespace mathworks

>oc adm policy add-scc-to-user privileged -z custom-sa --namespace mathworks
```

5.2 Prerequisite Software Checks

Complete the following software checks on the client machine before downloading and configuring MATLAB Online Server. The tasks can be done in any order.

- Make sure you can access the cluster’s kubeconfig file from the client machine.
- Make sure the client machine OS is supported. Refer to Supported platforms.
- Install Helm 3.
- Install Kubectl.
- Install Docker 18.09.

5.3 Step 1. Install Network License Manager

Install the network license manager for MATLAB on your license server. Configure the server to be accessed over the network.

- You can use an existing network license server or you can create a new installation. If you are performing a new license server installation, follow the instructions in Install Network License Manager with an Internet Connection on the MathWorks website. If your machine is not connected to the Internet, see Install Network License Manager Offline.

- The server and the MATLAB workers need access to their MathWorks licenses. Those licenses can be either co-located on the same license server or on independent servers. For more information, see License Manager Installation and Management on the MathWorks website.

- The MATLAB and server license host name and port(s) can be specified using the MLM_LICENSE_FILE environment variable format of <port>@<host>[,<port>@<other-host>]. The host and port must be accessible from inside the Kubernetes cluster.

- If the license server is configured correctly, then the license service running in the Kubernetes cluster runs without any errors and checks out the license when it starts. If the license is invalid or the license server cannot be reached, then the license service exits and Kubernetes automatically attempts to restart the license service periodically until it succeeds.
- The other services require that the license server is running, but they will continue to run if the license is temporarily unavailable. After an approximately two-hour grace period, the services go into a hibernation state and reject requests with a failed status. The logs of the services indicate when they enter or exit this hibernation or grace period state.

5.4 Step 2. Download MATLAB for Offline Installation

Download MATLAB for offline installation and install on the offline server.

Follow the Linux instructions for installing MATLAB on an offline machine. Make sure you install MATLAB into the folder /MATLAB.

5.5 Step 3. Download MATLAB Online Server Installer

Follow the instructions in Download MATLAB Online Server Installer in the Online Install on Single Machine instructions, and then return here and continue with these instructions.

5.6 Step 4. Configure Server

Edit the file `install.config`, located in the folder where you unzipped the installer. Consider creating a backup of `install.config` before editing to keep a copy of the default settings.

In the file, modify the fields in the following table.

Field Name	Description	Example
Virtual Machine Configuration		
`DOMAIN_BASE`	Domain name used to access the services. Provide the full domain name for this parameter, for example, matlab.mycompany.com. The domain you specify must be registered.	`DOMAIN_BASE=matlab.mycompany.com`

Authentication Configuration

ML_PASSWORD	MATLAB Online sign-in password The username for this bootstrapping mode is *admin*. NOTE: LDAP configuration is covered below.	ML_PASSWORD=password

License Configuration		
LICENSE_SERVER	License server details	You must have FLEXlm installed. Provide the port and the IP address of the license server, for example: LICENSE_SERVER=27000@172.0.0.1

If you have a remote registry within the company, configure the following information about the registry.

Remote Docker Registry Configuration		
DOCKER_REGISTRY	Remote Docker registry to push Docker images to	Registry name; for example: DOCKER_REGISTRY=myregistry.mycompany.com
DOCKER_REPOSITORY	Remote Docker repository	Repository name, for example: DOCKER_REPOSITORY=mos
IMAGE_PULL_SECRET	User-friendly name for the Kubernetes secret object which is created from registry credentials (when you deploy MATLAB Online Server). This secret allows the nodes to pull images from the remote registry.	Secret object name, for example: IMAGE_PULL_SECRET=mymossecret

5.7 Step 5. Security Configuration

Configure your environment with Transport Layer Security (TLS). See Security Configuration and apply the settings that are appropriate for your organization.

5.8 Step 6. Prepare and Push Docker Images

To allow cluster nodes to use Docker images, you must provide credentials to access the registry. If you have logged in to the remote registry from your client machine at least one time, then the file at ~/.docker/config.json has the registry credentials.

1. Copy the file with the following command:

   ```
   cp ~/.docker/config.json
   ./matlab_online_server/dockerconfig.json
   ```

   Contents from this file are used to create a Kubernetes secret by the name <IMAGE_PULL_SECRET> that is configured in Step 3.

2. Push Docker images to the remote registry (configured in Step 3) with the following command:

   ```
   ./mosadm push-docker-images
   ```

5.9 Step 7. Prepare Helm Charts and Configure Overrides

When the cluster and Docker images are ready to be used, the Helm charts and configuration overrides can be created before deploying the services. Helm is an open source project that helps deploy services into Kubernetes.

1. Issue the following command to copy the necessary Helm charts:
   ```
   ./mosadm copy-helm-charts
   ```

2. Enter the generate-overrides command with the flag --skip-matlab-image, as shown:

   ```
   ./mosadm generate-overrides --skip-matlab-image
   ```

   The mosadm command looks for the MATLAB Docker image locally even when the image is available in the remote Docker repository. So that mosadm can proceed without its having to download the image locally, passing the flag --skip-matlab-image is necessary.

3. Update the file `./overrides/<cluster>/<namespace>/all.yaml` to apply the following setting:

```
global:
  allowArbitraryUserId: true
```

MATLAB Online Server runs its containers with a specific UserId, whereas OpenShift randomizes the container's UserId based on a range that is different for different projects in OpenShift. You must set this property to true to allow OpenShift to choose an arbitrary UserId.

4. Update the file `./overrides/<cluster>/<namespace>/all.yaml` to add the ingress controller and its annotations.

   MATLAB Online Server supports Bring-Your-Own-Ingress, and OpenShift clusters come with an ingress controller. To use OpenShift's ingress controller, set the following fields:

   ```
   global:
     ingressController:
       name: openshift-ingress
       annotations:
            haproxy.router.openshift.io/timeout=365s
   ```

   MATLAB Online Server expects the timeout to be set to "365s," as the default timeout of openshift-ingress is not sufficient for MATLAB Online Server workflows.

5. Update the override file `./overrides/<cluster>/<namespace>/matlab-pool.yaml`.

   The changes you must implement are required for the following reasons:

   - The MATLAB pool pod needs to run with a specific UserId. Because the pod also runs a privileged container to set up user storage and proxy to MATLAB, it needs to use a custom service account that has access to "privileged" security context constraints.
   - The apparmor security profile, which is enabled by default, must be disabled since it is not supported on RedHat.

   ```
   # Allow use of custom service accounts
   customServiceAccount:
     enabled: true
     name: custom-sa
   apparmor: disabled
   ```

You can change the service account name in this example to the name you created for the account.

5.10 Step 8. Deploy MATLAB Online Server

1. After you finish customizing the overrides, execute the following command to install all the MATLAB Online Server components:

   ```
   ./mosadm deploy
   ```

   This command uses Helm to install all the charts, customized with any overrides that you generated in the previous steps.

2. Check that the services are all running with kubectl, replacing *your-namespace* with the namespace you used for MATLAB Online Server:

   ```
   sudo kubectl get pods --namespace your-namespace
   ```

The pods running MATLAB can take several minutes to fully start (4/4 containers). It is important to wait until the pods are in the ready state. If any of the containers do not start running, check the license server and the install.config settings. See MATLAB Pod Issues.

```
ubuntu@ubuntu:~/matlab_online_server$ kubectl get pods --namespace your-namespace

Output:

NAME                                                 READY    STATUS     RESTARTS   AGE

your-namespace-core-ui-core-ui-5bd9b7c88b-fqbl4      1/1      Running    0          111s

your-namespace-authnz-authnz-7c7999589c-x74d9        1/1      Running    0          111s

your-namespace-gateway-669dd7865f-t6l74              1/1      Running    0          110s

your-namespace-license-548f7b6655-8skdt              1/1      Running    0          108s

your-namespace-matlab-pool-8c679f76-c4h4n            4/4      Running    0          107s

your-namespace-matlab-pool-8c679f76-cckkp            4/4      Running    0          107s

your-namespace-matlab-pool-ui-579fc8ff49-2bz25       1/1      Running    0          107s

your-namespace-resource-resource-894447df7-r9ngb     1/1      Running    0          103s
```

5.11 Step 9. Connect to MATLAB Online

After you have installed MATLAB Online Server successfully, provide a way to access MATLAB Online from the browser.

The OpenShift ingress controller creates a service object to access the cluster from outside. To get the service, issue the following command:

```
oc get services --namespace openshift-ingress
```

This command shows an IP address in the External IP column. Using the DNS service of your choice (for example, Route53), map `DOMAIN_BASE` to this IP address.

5.12 Step 10. Verify MATLAB Online Server

Make sure the address in the DOMAIN_BASE parameter defined in install.config is registered and can be routed to the IP address of the host machine.

```
ping -c 1 matlab.domain.com
```

This command returns the IP address of the host machine. In some types of installation, the ping does not receive packets. If you have such an installation, this command returns the IP address of the server you ping.

Open a browser on another machine and go to the address where you are hosting MATLAB Online:

matlab.domain.com/matlabonline

The address is the `DOMAIN_BASE` parameter defined in `install.config`. Your browser window shows a prompt to select a password only. For a single-machine installation, use the user name `admin` and the password `password`. Normally, the password is the value set in the `ML_PASSWORD` parameter (defined in `install.config`).

If you can now access your internal version of MATLAB Online, installation is complete. If you cannot access MATLAB Online, or you run into any other technical error, contact MathWorks Support.

5.13 Next Steps

If you want to apply additional configuration parameters, see Post-Installation Additional Configuration.

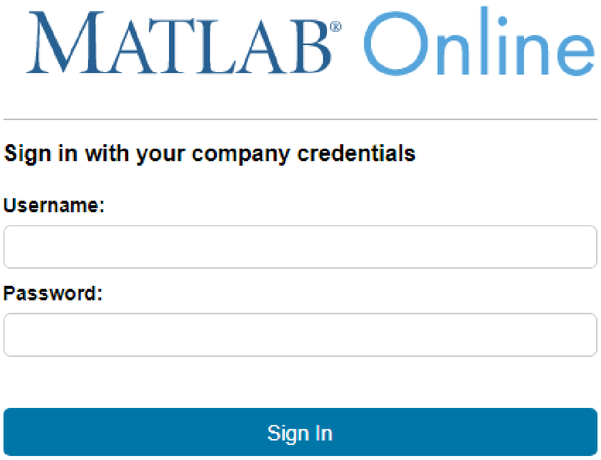
6 Post-Installation Additional Configuration

6.1 Overview

The default configuration parameters allow a single-machine installation. These configuration settings are stored in a collection of YAML files. The following additional configurations allow customization of the system for one or multiple machines.

6.2 Customize Organization Name and Logo

MATLAB Online Server users can access MATLAB over the web via MATLAB Online as soon as they sign in with the credentials that are configured in LDAP and Authorization. The default sign-in page for MATLAB Online, with no customization, is shown in the following image:



Sign in with your company credentials

You can customize the appearance of this sign-in page by specifying your organization name and logo.

6.2.1 Specify Organization Name

To specify your organization name, edit the settings in the file all.yaml and then add the organizationName attribute to the global config file.

For example:

```
global:
  organizationName: "MathWorks"
```

6.2.2 Specify Logo

To specify your organization logo, you can choose either to link to the logo or use a logo file.

6.2.2.1 Link to Logo

Edit the core-ui.yaml file and then add the following configuration setting:

```
loginPage:
  logo:
      url: <url to the logo>
```

6.2.2.2 Use Logo File

1. Copy or rename the logo file to loginPage.logo.file.
2. Copy this file to the following location before deploying the services:

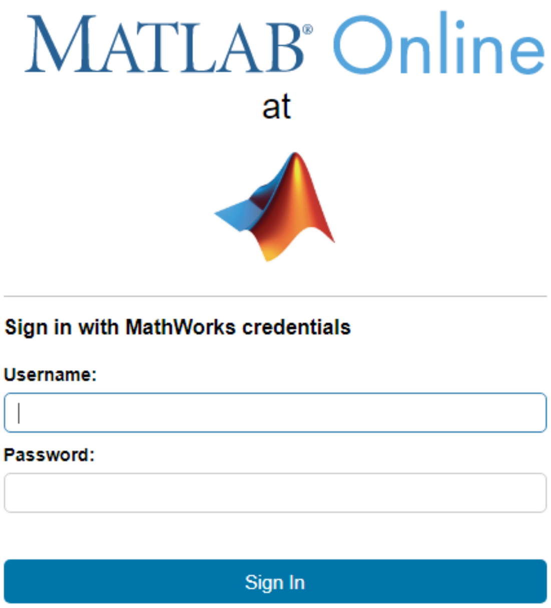
   <Overrides_dir>/<cluster>/<namespace>/core-ui/binary/loginPage.logo.file

6.2.3 Implement Customized Sign-In Page

To implement the new sign-in page, redeploy MATLAB Online Server using the following command:

```
./mosadm upgrade core-ui
```

The next time you start MATLAB Online from MATLAB Online Server, the sign-in page shows your organization name and logo. The following image is an example of a customized sign-in page for MathWorks MATLAB Online on MATLAB Online Server:



6.3 LDAP and Authorization

Lightweight Directory Access Protocol, also known as LDAP, is used to manage user identities. Identity management for MATLAB Online Server can apply user information from sources that you specify. LDAP configuration provides access control and tells MATLAB Online Server how to map user information tables to user information that MATLAB Online Server needs to manage these users.

To apply a custom LDAP configuration, edit the settings in the authnz.yaml file. The parameters are summarized in the following sections. For more information about LDAP settings, see https://ldap.com/.

For information on local settings, see Local Accounts.

6.3.1 LDAP Settings

To apply LDAP settings, set enabled to true.

If you enable LDAP, then you must also configure the following LDAP settings:

- host
- port
- baseDn
- adminDn
- adminPassword

The other settings are optional.

6.3.1.1 Enable LDAP

```
ldap:
 enabled:
```

Boolean setting to indicate whether to use an enterprise LDAP in place of local accounts. Use of LDAP and local accounts is mutually exclusive. Set enabled to either true or false. If enabled is false, the other LDAP settings do not apply.

6.3.1.2 host

```
ldap:
  host:
```

Host address for LDAP, for example, `ldapserver.yourcompany.com`. If LDAP enabled is set to `true`, you must configure the host attribute.

6.3.1.3 port

```
ldap:
  port:
```

LDAP port at the host address, for example, 389 for regular LDAP and 636 for secure LDAP. If LDAP enabled is set to `true`, you must configure the port attribute.

6.3.1.4 baseDn

```
ldap:
  baseDn:
```

Starting point for the search tree in your LDAP environment. For example, assuming baseDn = dc: dc=ldap,dc=yourcompany,dc=com. If LDAP enabled is set to true, you must configure the baseDN attribute.

6.3.1.5 filter

```
ldap:
  filter:
```

LDAP search filter. Specifying a search filter to scan a subset of users from the LDAP database allows an administrator to reduce the scope of the LDAP search and control the organizational users that can access MATLAB Online Server. You can leverage variable substitution to configure how the user name is provided to LDAP, for example, "`(CN={0})`".

Specify the search filter as attribute=value. For instance, CN=test* matches all users that have a common name (CN) attribute that starts with "test". Use parentheses to combine multiple filter expressions in an AND (&) or OR(|) clause. For example, `(&(CN={0})(department=foo))` matches all users that are in department "foo".

The default search filter is objectClass=organizationalPerson. For more information on search filters, see the LDAP Filters website.

6.3.1.6 adminDn

```
ldap:
  adminDn:
```

Administrator account used to establish the connection to the LDAP server, for example, `cn=readonly`. This account is used to query LDAP for the list of users. The account requires only read access to the directory and does not attempt to store information back into LDAP.

6.3.1.7 adminPassword

```
ldap:
  adminPassword:
```

Administrator account password used to establish the connection to the LDAP server.

6.3.2 Secure LDAP Settings

Whether you need to set secure LDAP settings depends on your organization’s setup.

6.3.2.1 useSsl

```
ldap:
  useSsl:
```

Boolean flag to indicate if your LDAP server requires SSL for connection. Set to either true or false.

6.3.2.2 useStartTLS

```
ldap:
  useStartTLS:
```

Boolean flag to indicate if your LDAP server requires start TLS for connection. Set to either true or false.

6.3.2.3 validateServerCertificate

```
ldap:
  validateServerCertificate:
```

Boolean flag to specify whether MATLAB Online Server uses the serverCertificate token configuration. Set to either true or false.

6.3.2.4 serverCertificate

```
ldap:
  serverCertificate:
```

Text attribute containing the server certificate if your server requires a client token to establish a secure connection.

6.3.3 Subject Attribute Mappings

The attribute names in your LDAP can differ from what MATLAB Online Server expects. The subject attribute mapping lets you set values for the user attributes that map to your organizational LDAP values.

For example, your user's first name might be stored as *displayFirstName* in the LDAP system. To map this in MATLAB Online Server, set `displayName: displayFirstName`.

6.3.3.1 Subject Map

```
ldap:
  subjectAttributeMapping:
    subjectId:
    displayName:
    extra:
        uid:
        guid:
```

displayName is an optional attribute that controls what is displayed for the user name in the upper-right-hand corner of MATLAB Online Server.

subjectId is the default mapping attribute for the user account, normally `cn`.

The extra parameters are for additional customization relating to the network files system (NFS). There are two common settings:

- extra.uid — Mapping to a Unix UID
- extra.gid — Mapping to the Unix Group ID

These settings are required for storage profiles.

6.3.4 Local Accounts

You can choose to use local accounts in place of an LDAP system. Use of local accounts and LDAP is mutually exclusive.

6.3.4.1 Enable Local Accounts

```
local:
  enabled:
  accounts:
```

Enable local accounts by setting to true. This setting is mutually exclusive with LDAP.

`accounts` contains the hard-coded account information for local users. The format for this line is the user name and user password separated by a pipe delimiter |. Multiple accounts can be defined here as a comma-separated list. For example:

```
username1|userpassword1,username2|userpassword2
```

6.3.5 SAML and Authorization

Identity management for MATLAB Online Server lets you apply user information from sources that you specify. MATLAB Online Server supports integration via the Security Assertion Markup Language, also known as SAML. The SAML configuration for MATLAB Online Server ensures that the identities used to control access to MATLAB Online Server are managed by the enterprise Identity Provider.

To apply a custom SAML configuration, edit the settings in the authnz.yaml file. The parameters are summarized in the following sections. For more information about SAML settings see https://wiki.oasis-open.org/security/ .

6.3.5.1 SAML Settings

To enable the use of a SAML integration, set enabled to true.

If you enable SAML, then you must also configure the following SAML settings:

- corsalloworigindomain
- relyingPartyId
- idpMetadataXml

6.3.5.1.1 Enable SAML

```
saml:

  enabled: true
```

Boolean setting to indicate whether an enterprise SAML integration is being used in place of local accounts. Use of SAML and local accounts is mutually exclusive. Set enabled to either true or false. If enabled is false, the other SAML settings do not apply.

6.3.5.1.2 corsalloworigindomain

```
saml:

  corsalloworigindomain:
```

corsalloworigindomain is the host name of the server that communicates with MATLAB Online Server, for example, samlintegrationhost.yourcompanydomain.com. If SAML enabled is set to true, you must configure the corsalloworigindomain attribute.

6.3.5.1.3 relyingPartyId

```
saml:

  relyingPartyId:
```

According to the SAML specification, the value of relyingPartyId specifies the hostname for the server that integrates with the SAML Identity Provider. Consequently, this value would be the host name of the dns entry for your MATLAB Online Server kubernetes cluster.

However, depending on the integration product or service being used to integrate with the Identity Provider via the SAML, this may be an arbitrary identifier. If SAML enabled is set to true, you must configure the relyingPartyId attribute.

6.3.5.1.4 IdpMetadataXml

```
saml:

  idpMetadataXml:
```

`idpMetadataXml` is a block of xml that is exported by the SAML Identity Provider. This block of xml can simply be pasted in to an `authnz.yaml` file. However, note that the xml must be properly formatted YAML, otherwise you will encounter problems deploying `authnz` via the `mosadm` utility. If SAML enabled is set to `true`, you must configure the `idpMetadataXml` attribute.

6.3.5.2 Subject Attribute Mappings

The attribute names in your SAML assertions can differ from what MATLAB Online Server expects. The subject attribute mapping lets you set values for the user attributes that map to your organizational SAML assertion values.

For example, an end user’s first name might be stored as *displayFirstName* in the SAML system. To map this in MATLAB Online Server, set displayName: displayFirstName.

6.3.5.2.1 Subject Map

```
saml:

  subjectAttributeMapping:

    displayName:
```

```
    groups:

    extra:

        uid:

        guid:
```

`displayName` is an optional attribute that controls what is displayed for the user name in the upper-right-hand corner of MATLAB Online Server.

`groups` is an optional attribute that can be used for additional customization relating to the network file system (NFS). See the subsection on Placeholders in the Storage Profiles section.

The extra parameters are for additional customization relating to the network files system (NFS). There are two common settings:

- `extra.uid` — Mapping to a Unix UID
- `extra.gid` — Mapping to the Unix Group ID

These settings are required for storage profiles.

6.4 MATLAB Pool

MATLAB Pool (matlab-pool.yaml) settings determine how individual users can interact with the online server. There are individual and group settings for users.

6.4.1 Available MATLAB Pool Pods

```
replicaCount: 2
```

replicaCount determines the number of available MATLAB Pool pods kept available and ready to be used that can be assigned to a user upon login. This count is to make sure that users can sign in within a few seconds. If the number of new logins exceeds this count, the user is delayed while a pod is started. The default value is 2.

The maximum number of active users is limited to the number of MATLAB licenses. The total number of MATLAB instances is limited to the number of licenses and the hardware configured for the MATLAB Pool.

6.4.2 License Servers

```
flexlm:
  servers:
```

License server information for checking out MATLAB licenses. flexlm (FlexNet Publisher) is the network license manager used by MathWorks.

List server information in the format port:address, for example, "27000:flexlm.mycompany.com". You can list information for multiple servers separated by a colon, for example:

```
  servers: "27000@flexlm1.company.com:27000@ flexlm2.company.com"
```

6.4.3 Memory and CPU Configuration

The default memory and CPU configuration for the MATLAB container contains the following settings:

```
matlabResources:
  requests:
    cpu: 200m
    memory: 2Gi
  limits:
    #cpu:
    #memory:
```

The default settings allow the MATLAB container to request a minimum of 200m CPU and 2Gi of memory on the node, but there is no limit set on these resources. With the default configuration, it is possible for a container to use up all CPU and memory on the node.

To change the configuration and put a limit on requests for CPU and memory, edit the limits section in matlab-pool.yaml to remove the commented-out lines (#) and then add values that are appropriate for your environment and infrastructure. For example:

```
matlabResources:
  requests:
    cpu: 200m
    memory: 2Gi
  limits:
    cpu: 500m
    memory: 7Gi
```

With these settings, the MATLAB container requests a minimum of 200m CPU and 2Gi of memory and can go only to a *maximum* of 500m CPU and 7Gi memory. Therefore, the container cannot use the resources on the node completely.

After updating the matlab-pool setting, issue the following command to upgrade the MATLAB Pool:

```
./mosadm upgrade matlab-pool
```

Note: For more about Kubernetes and CPU and memory, consult the following resources:

- CPU: https://kubernetes.io/docs/concepts/configuration/manage-compute-resources-container/#meaning-of-cpu
- Memory: https://kubernetes.io/docs/concepts/configuration/manage-compute-resources-container/#meaning-of-memory

6.4.4 Configure Resource Timeouts

To configure resource timeouts, you can uncomment and set required values in the matlab-pool.yaml override.

```
# Can be used to set arbitrary annotations on the pod
extraAttributes:
  # Default timeouts are listed. Uncomment and set required values
    #minLifetimeMinutes: 5
    #maxLifetimeMinutes: 240
    #defaultLifetimeMinutes: 120
    #minInactivityTimeoutMinutes: 5
    #maxInactivityTimeoutMinutes: 30
    #defaultInactivityTimeoutMinutes: 15
```

After updating the matlab-pool setting, issue the following command to upgrade the MATLAB Pool:

```
./mosadm upgrade matlab-pool
```

6.4.5 Environment Variables

Environment variables required for MATLAB. List all the environment variables under `matlabProcessEnv`. These variables are set for the MATLAB process.

Use the format `<ENV_VAR_NAME>: <ENV_VAR_VALUE>`. For example:

```
matlabProcessEnv:
#Set the environment variables for MATLAB here
#Ex: PCT_ENABLED: "true"
```

6.4.6 Storage Profiles

A storage profile is a specification defined by the MATLAB Online Server administrator that determines how and which drives, directories, and files are available to the resource (MATLAB) after users sign on.

The storage profiles are defined in the MATLAB Pool configuration:
<OVERRIDES_DIR>/<cluster>/<namespace>/matlab-pool.yaml

The storage profiles are mounted when the user signs in and the remote resource (MATLAB) is allocated to them.

The login information provides more data regarding the user at the time of mounting the profile to the Storage service (part of the MATLAB pod). This information is used to resolve and mount user-specific directories.

Storage profiles allow you to specify this dynamic configuration through the use of *placeholders*. These placeholders are resolved dynamically while the directories are mounted with the same information from a user who has signed in (from the AuthNZ service). For more information about placeholders, see Placeholders.

6.4.6.1 Profile Specification

Attribute	Default Value	Optional	Note
`name`	`""`	No	Name of the storage profile.
`startDirectory`	`"/"`	Yes	Start directory (user path of the MATLAB user) after the storage profile is mounted.
`restrictNavigationOutsideMounts`	`false`	Yes	If `true`, MATLAB users are not able to navigate (cd) to other directories other than those specified in mounts and MATLAB directory.
`mounts`	`<Nil>`	Yes	Directories to be mounted and made available to the MATLAB user (see Mount Specifications).
`addons`	`<Nil>`	Yes	Directory where MATLAB add-ons are stored (see Add-ons Specification).

In YAML format:

```
storage:
  profiles:
    - name: <name>
      mounts:
              <list of mounts>
      startDirectory: <start directory>
      restrictNavigationOutsideMounts: <true/false>
      addons:
        directory: <addons directory>
```

6.4.6.2 Specification for Add-Ons

Attribute	Default Value	Optional	Note

`directory`	`<startDirectory>/MATLAB Add-Ons`	Yes	Directory where MATLAB add-ons are stored.

6.4.6.3 Mount Specifications

Attribute	Default Value	Optional	Note
`name`	`""`	No	Name of the mount.
`mountPath`	`""`	No	Path of the mount. This info is available to the MATLAB user as the mounted path, after the user signs in.
`type`	`""`	No	Type of the mount. Based on the type, the mount properties can vary. For more information, see Mount Types.
`permissionType`	`""`	No	Type of the permissions that are given to the mounted directory after the user has signed in. Based on the permission type, the mount properties can vary. For more information, see Mount Types.

6.4.6.4 Mount Types

MATLAB Online Server supports the following mount types: Empty, Host, and NFS.

6.4.6.4.1 Empty

The Empty (**empty**) mount type creates an empty directory. This storage is ephemeral storage, meaning that after a user signs out of the session, all data inside the directory is deleted permanently.

This mount type does not require any extra properties.

In YAML format:

```
storage:
  profiles:
    - name: <name>
      mounts:
        - name: code
          mountPath: /code
          type: empty
          permissionType: <permission type>
      startDirectory: <start directory>
      restrictNavigationOutsideMounts: <true/false>
      addons:
        directory: <addons directory>
```

6.4.6.4.2 Host

The Host (**host**) mount type mounts a directory from the Nodes host. This mount type expects the directory (**hostPath**) to be present on the Node; otherwise, the MATLAB pod does not start.

This mount type requires the following extra property:

- hostPath

In YAML format:

```
storage:
  profiles:
    - name: <name>
      mounts:
        - name: node_folder
          mountPath: /share
          hostPath: /share
          type: host
          permissionType: <permission type>
      startDirectory: <start directory>
      restrictNavigationOutsideMounts: <true/false>
      addons:
        directory: <addons directory>
```

6.4.6.4.3 NFS

The NFS (**nfs**) mount type mounts an NFS directory and makes it available to the MATLAB user.

This mount type expects that the NFS server allows mounting the NFS directories onto the MATLAB pods. The NFS administrator might need to allow access to the NFS server by the Kubernetes cluster, through the NFS exports list.

This mount type requires the following extra properties:

- server
- subPath
- mountOptions

In YAML format:

```
storage:
  profiles:
    - name: <name>
      mounts:
        - name: nfsHome
          mountPath: /share
          type: nfs
          server: "nfs"
          subPath: "/exports/home/${subject.subjectId}"
          mountOptions:
"rw,relatime,vers=3,rsize=1048576,wsize=1048576,namlen=255,acregmin=60
0,acregmax=600,acdirmin=600,acdirmax=600,hard,nocto,noacl,proto=tcp,ti
meo=600,retrans=2,mountproto=tcp,local_lock=none,nolock"
          permissionType: <permission type>
      startDirectory: <start directory>
      restrictNavigationOutsideMounts: <true/false>
      addons:
        directory: <addons directory>
```

6.4.6.5 Permission Types

The following permission types are supported: None, Fixed, and User.

6.4.6.5.1 None

The None (**none**) permission type performs a mount with default access to the MATLAB process only.

This permission type is ideal for empty mount types (ephemeral storage) or anonymous read-only directories that are mounted.

This permission type does not require any extra properties.

In YAML format:

```
storage:
  profiles:
    - name: <name>
      mounts:
        - name: code
          mountPath: /code
          type: empty
          permissionType: none
      startDirectory: <start directory>
      restrictNavigationOutsideMounts: <true/false>
      addons:
        directory: <addons directory>
```

6.4.6.5.2 Fixed

The Fixed (**fixed**) permission type performs a mount and shares the directory with the same permissions that are specified (**uid**, **gid**) to the MATLAB process only.

This permission type is ideal for shared read-only directories.

This permission type requires the following extra properties:

- uid
- gid

In YAML format:

```
storage:
  profiles:
    - name: <name>
      mounts:
        - name: public
          mountPath: /public
          type: nfs
          server: "nfs"
          subPath: "/exports/nfs_share"
          mountOptions:
"ro,relatime,vers=3,rsize=1048576,wsize=1048576,namlen=255,acregmin=60
0,acregmax=600,acdirmin=600,acdirmax=600,hard,nocto,noacl,proto=tcp,ti
meo=600,retrans=2,sec=sys,mountproto=tcp,local_lock=none,nolock"
          permissionType: fixed
          uid: 1001
          gid: 1000
      startDirectory: <start directory>
      restrictNavigationOutsideMounts: <true/false>
      addons:
        directory: <addons directory>
```

6.4.6.5.3 User

The User (**user**) permission type performs a mount and shares the directory with the same permissions as that of the user (**uid**, **gid**) to the MATLAB process only if the placeholders are resolved. For more information, see Placeholders.

This permission type is ideal for mounting user-specific directories, such as an NFS home folder, which is readable and writable by the signed-in user only.

This permission type requires the following extra properties:

- uid
- gid

In YAML format:

```
storage:
  profiles:
    - name: <name>
      mounts:
        - name: public
          mountPath: /public
          type: nfs
          server: "nfs"
          subPath: "/exports/home/${subject.subjectId}"
          mountOptions:
"rw,relatime,vers=3,rsize=1048576,wsize=1048576,namlen=255,acregmin=600,acregmax=600,acdirmin=600,acdirmax=600,hard,nocto,noacl,proto=tcp,timeo=600,retrans=2,mountproto=tcp,local_lock=none,nolock"
          permissionType: user
          uid: "${subject.uid}"
          gid: "${subject.gid}"
      startDirectory: <start directory>
      restrictNavigationOutsideMounts: <true/false>
      addons:
        directory: <addons directory>
```

6.4.6.6 Disallow UIDs

MATLAB Online Server includes a security feature that disallows certain UIDs to create storage mounts.

You can add a configurable list of UIDs that are not allowed to set up the storage permissions. For example, if a user with uid 0 signs in, the permissions on the binded folder are given the root permissions (as uid = 0); this feature disallows this permission binding for uid 0.

The following shows the default list of disallowed Uids (in YAML format); you can override the values with your specific disallowed Uids:

```
storage:
  disallowedUids:
    - "0"
    - "1000600715"
    - "1000600903"
    - "1000600912"
```

6.4.6.7 Placeholders

Placeholders are temporary markers for values that are resolved dynamically while the directories are mounted during storage setup.

For more information on the AuthNZ IdP (Identify Provider) configuration and the Subject response, see LDAP and Authorization.

A sample response from AuthNZ follows. In this case, with the same information for the signed-in user from the AuthNZ service.

```
{
   "mwtype":"authnz/SubjectAuthResponse",
   "subject":{
      "subjectId":"username1",
      "displayName":"username1",
      "groups":[

      ],
      "extra":{
         "uid":[
            "3904"
         ],
         "gid":[
            "3904"
         ]
      },
      "idp":"ldap"
   },
   "authToken":"xxxxxxx-xxxxxxxx-xxxxxx",
   "authenticated":true
}
```

From the above response, the following placeholders can be resolved:

- subject.subjectId
- subject.displayName
- subject.groups
- subject.uid
- subject.gid

As another example, consider a storage unit with one NFS directory per user that must be mounted, where:

- Each user has a different home directory that is differentiated by the user name.
- The home directory of each user is locked to their UID and GID.

The storage profile can be constructed as follows:

```
storage:
  profiles:
    - name: nfsHome
      startDirectory: "/home/${subject.subjectId}"
      mounts:
        - name: home
          permissionType: user
          mountPath: "/home/${subject.subjectId}"
          uid: "${subject.uid}"
          gid: "${subject.gid}"
          type: nfs
          server: "nfs"
          subPath: "/exports/home/${subject.subjectId}"
          mountOptions:
"rw,relatime,vers=3,rsize=1048576,wsize=1048576,namlen=255,acregmin=600
,acregmax=600,acdirmin=600,acdirmax=600,hard,nocto,noacl,proto=tcp,time
o=600,retrans=2,mountproto=tcp,local_lock=none,nolock"
      restrictNavigationOutsideMounts: false
```

Note that mountPath cannot be started with a placeholder.

6.5 Multiple MATLAB Pools

6.5.1 Overview

Multiple pool support in MATLAB Online Server allows you to provide an interface that enables end users to choose the MATLAB experience they want, for example, MATLAB R2020a or R2020b, or R2020b with GPU access.

The multiple chooser UI, which is the interface the end user sees when they first sign in to MATLAB Online, is enabled by configuring a core-ui.yaml override.

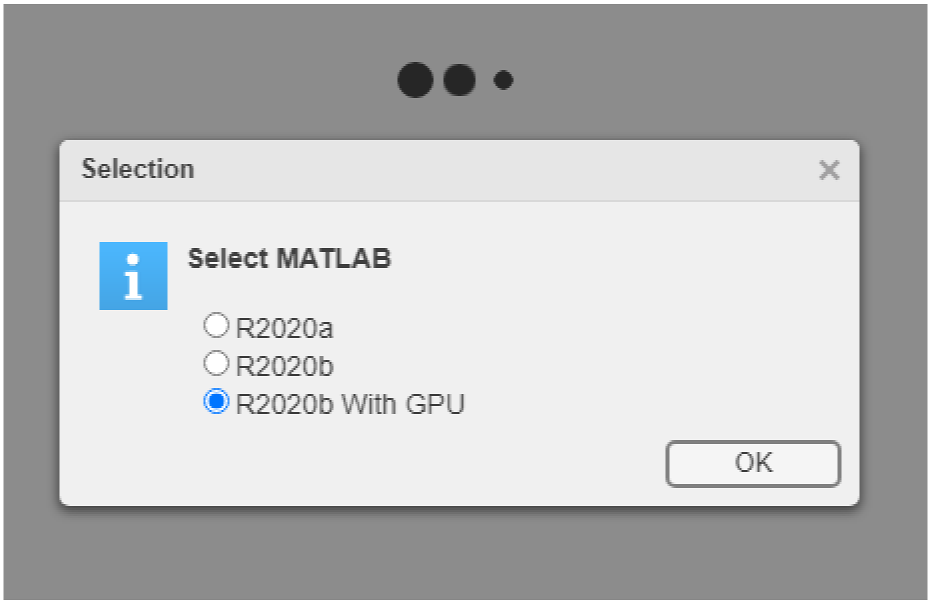
6.5.2 Enable Multiple Pool Chooser

To enable the multiple chooser UI, add the following core-ui.yaml override.

```
multiplePoolSupport:
  enabled: true
```

After you successfully deploy MATLAB Online Server with this setting and all the Kubernetes pods have started successfully (see substeps 2 and 3 in Step 4. Install Kubernetes), MATLAB Online displays the multiple chooser UI to the end user after they sign on.

The options end users can choose from depend on how many and what kind of pools you create. In the following example, the administrator created three pools, allowing the end user to select from three different versions of MATLAB Online:



Selection

Select MATLAB

○ R2020a

○ R2020b

● R2020b With GPU

When the end user clicks **OK**, MATLAB Online Server launches the selected version of MATLAB in the browser.

6.5.3 Create Multiple Pools

6.5.3.1 First MATLAB Pool

For the first pool with the MATLAB you installed with MATLAB Online Server installation, perform the following steps:

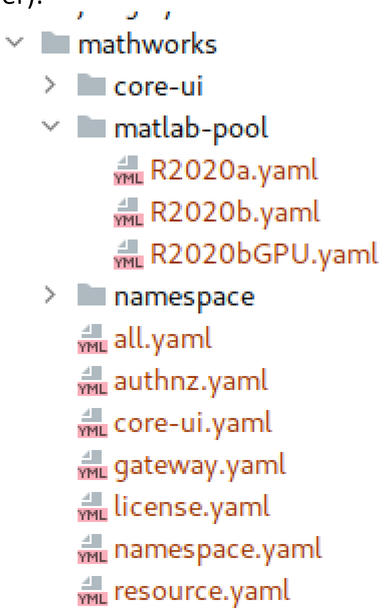
1. Create a directory named matlab-pool.
2. Move the file matlab-pool.yaml to the matlab-pool directory.
3. Rename the matlab-pool/matlab-pool.yaml to matlab-pool/RXXXXy.yaml, where XXXX is the release year and y is the release version (a or b) of MATLAB, along with any additional characters to describe the pool, for example, R2020a, R2020b, R2020bGPU, and so on.
4. (Optional) Configure the MATLAB pool (see MATLAB Pool).
5. Configure additional pool options (see Configure Additional Pool Options).

6.5.3.2 Additional MATLAB Pools

1. On the server, install the version of MATLAB you want to create a pool for.
2. Copy the yaml file you created for the first MATLAB version (for example, R2020a.yaml) to matlab-pool/RXXXXy.yaml, where RXXXXy is the release version of the additional MATLAB you installed.
3. (Optional) Configure the new MATLAB pool (see MATLAB Pool).
4. Configure additional pool options for the new MATLAB pool (see Configure Additional Pool Options).

Repeat the above steps for each MATLAB pool you want to add.

After you have created the pools, the override file tree looks similar to the following (your names might differ):



```
mathworks
  core-ui
  matlab-pool
    R2020a.yaml
    R2020b.yaml
    R2020bGPU.yaml
  namespace
  all.yaml
  authnz.yaml
  core-ui.yaml
  gateway.yaml
  license.yaml
  namespace.yaml
  resource.yaml
```

6.5.4 Configure Additional Pool Options

For each pool yaml file that you create, you must configure the MATLAB pool for the UI. Add the following configuration to the RXXXXy.yaml file, for example, for R2020a.yaml:

```
poolConfig:
  configId: R2020a
  displayName: "R2020a"
```

The following values must be unique for each MATLAB pool you create.

Property	Default Value	Description
`configId`	Release name and number, for example, R2020a	A unique identifier for each pool
`displayName`	Release name and number, for example, R2020a	The display name is the name shown as an option in the multiple chooser UI. After the MATLAB session is established, the display name appears as the title of the browser tab.

6.6 Specify MATLAB Documentation

Specify which help content serves all the MATLAB pools you are providing:

1. Select the version of MATLAB (for example, R2020b) that you want to serve the help content. Open the respective override (R2020b.yaml), and add the following:

   ```
   includeHelpSearch: true
   ```

2. Configure all MATLAB pools to use the help content from the selected version. To the file all.yaml, add the following configuration:

   ```
   extraMatlabStartupFlags: -r
   "docRelease=com.mathworks.mlservices.MLHelpServices.setDocRelea
   se('R2020b');docRelease=com.mathworks.mlservices.MLHelpServices
   .getDocRelease;clear docRelease;"

   includeHelpSearch: false
   ```

   Replace ‘R2020b’ with the MATLAB version you are using for the help content.

6.7 Redeploy MATLAB Online Server

After you complete the overrides to provide multiple MATLAB pools, redeploy MATLAB Online Server with the following commands:

1. Undeploy MATLAB Online Server

   ```
   ./mosadm undeploy
   ```

2. Deploy MATLAB Online Server

   ```
   ./mosadm deploy
   ```

When end users next sign on, they will be able to choose from the different MATLAB versions you have set up for them.

Part 2: Operations

7 Administration

7.1 Manage MATLAB Online Server

Manage services and images for MATLAB Online Server using mosadm. The syntax for mosadm is as follows:

```
mosadm [options...] <operation> [service]
```

Use mosadm to perform the following tasks:

- Load, list, and push Docker images.
- Build MATLAB images.
- Deploy, upgrade, and undeploy services.
- Get system information.

7.1.1 Manage Docker Images

Load, list, and push Docker images.

7.1.1.1 Load Docker Image

Load Docker images from the installer into the local Docker cache.

7.1.1.1.1 Syntax

```
mosadm load-docker-images
```

7.1.1.1.2 Example Output

```
ubuntu@ip-172-31-67-165:~/matlab_online_server$ sudo ./mosadm load-docker-images

Will load docker images from ./data into local docker cache

Is the information displayed above correct (y/n)? y

Start loading images

…

Done loading images
```

7.1.1.2 List Docker Images

List all the Docker images in the local Docker cache.

7.1.1.2.1 Syntax

```
mosadm list-docker-images
```

7.1.1.2.2 Example Output

```
mathworks/mos/com.mathworks.matlabonlineserver.matlab-image:R2020a
mathworks/mos/com.mathworks.matlabonlineserver.matlab-omnibus-debian-
image:R2020a
mathworks/mos/com.mathworks.matlabonlineserver.omnibus-debian-image:R2020a
mathworks/mos/com.mathworks.matlabonlineserver.license-debian-image:R2020a
mathworks/mos/com.mathworks.matlabonlineserver.core-ui-debian-image:R2020a
mathworks/mos/com.mathworks.matlabonlineserver.gateway-image:R2020a
mathworks/mos/com.mathworks.matlabonlineserver.resource-image:R2020a
mathworks/mos/com.mathworks.matlabonlineserver.resource-proxy-image:R2020a
```

7.1.1.3 Push Docker Images

Retag and push the loaded Docker images to the configured remote registry. The mosadm command uses the Docker registry and Docker repository information from the install.config file.

7.1.1.3.1 Syntax

```
mosadm push-docker-images
```

7.1.1.3.2 Description

1. Update the following lines in the install.config file:

```
# Docker registry and repository, if the cluster is remote then
override this

DOCKER_REGISTRY=<docker-registry-here>

DOCKER_REPOSITORY=<repo-here>
```

2. Run the following command to push the Docker images:

```
mosadm push-docker-images
```

Alternatively, you can pass the command as a configuration option as follows:

```
mosadm push-docker-images --docker-registry name_of_docker-registry --
docker-repository name_of_repo
```

7.1.1.3.3 Example Output

```
ubuntu@ip-172-31-67-165:~/matlab_online_server$ ./mosadm push-docker-
images --docker-registry docker-registry.com/repo-prefix --docker-
repository mos


Will push docker images from mathworks/mos into docker-
registry.com/repo-prefix/mos

Note: this may take a while.

Is the information displayed above correct (y/n)? y

Pushing loaded images
```

```
docker tag mathworks/mos/com.mathworks.matlabonlineserver.matlab-
image:R2020a docker-registry.com/repo-
prefix/mos/com.mathworks.matlabonlineserver.matlab-image:R2020a

docker push docker-registry.com/repo-
prefix/mos/com.mathworks.matlabonlineserver.matlab-image:R2020a

The push refers to repository [docker-registry.com/repo-
prefix/mos/com.mathworks.matlabonlineserver.matlab-image]

…
```

7.1.2 Build MATLAB Image

Build an image containing MATLAB from /MATLAB.

7.1.2.1 Syntax

```
mosadm build-matlab-image MATLAB_install_location
```

7.1.2.2 Example Output

```
Will build the MATLAB image from base image from mathworks/mos

Note: this may take a while depending on the size of the installation
(~30 minutes)

Splitting the directories and files into layers

2020-01-17T17:47:24-05:00

Building layer 1/14, size: 2047 MB

Saving layer 1/14

sha256:0720c6fa30b6532974140e109f9e0360e9af4cf9af93ac6d204bee2481a6c6b
9

large-image-builder

2020-01-17T17:48:27-05:00

Layer 1/14 built

Building layer 2/14, size: 2047 MB

Saving layer 2/14

sha256:3def939b2f1f4568e620836aa0756916c6819a76ff4accee74a3cc75b018f07
6

…
```

7.1.3 Deploy, Upgrade, and Undeploy Services

Deploy the service into the Kubernetes cluster. <service> defaults to all charts with overrides.

7.1.3.1 Deploy Service

7.1.3.1.1 Syntax

```
    mosadm deploy [<service>]
```

7.1.3.1.2 Description

To deploy all services, namely authnz, core-ui, gateway, license, matlab-pool, namespace, nginx-ingress, and resource, use this syntax:

```
mosadm deploy
```

To deploy a specific service, add the service name:

```
mosadm deploy authnz

mosadm deploy matlab-pool
```

7.1.3.1.3 Example Output

```
ubuntu@ip-172-31-81-54:~/matlab_online_server$./mosadm deploy

Will perform deploy

Charts to deploy in matlab-online-server/mathworks:

authnz core-ui gateway license matlab-pool namespace nginx-ingress
resource

Is the information displayed above correct (y/n)? y

Found deployments for authnz: default

Deploying mathworks-authnz

helm --kube-context kubernetes-admin@matlab-online-server install --
namespace mathworks --name mathworks-authnz charts/authnz-chart -f
overrides/matlab-online-server/mathworks/all.yaml -f overrides/matlab-
online-server/mathworks/authnz.yaml

NAME:   mathworks-authnz

LAST DEPLOYED: Fri Jan 24 20:29:48 2020

NAMESPACE: mathworks

STATUS: DEPLOYED

RESOURCES:

==> v1/ConfigMap
```

```
NAME                     DATA  AGE

mathworks-authnz-authnz  2     0s

==> v1/Deployment

NAME                     READY  UP-TO-DATE  AVAILABLE  AGE

mathworks-authnz-authnz  0/1    1           0          0s

==> v1/NetworkPolicy

NAME                     AGE

mathworks-authnz-authnz  0s

==> v1/Pod(related)

NAME                                    READY  STATUS             RES
TARTS  AGE

mathworks-authnz-authnz-569df55f78-
6bdg4  0/1    ContainerCreating  0         0s

==> v1/Service

NAME    TYPE       CLUSTER-IP    EXTERNAL-IP  PORT(S)  AGE

authnz  ClusterIP  10.108.12.41  <none>       80/TCP   0s

==> v1beta1/Ingress

NAME    AGE

authnz  0s

…

Chart deploy complete
```

7.1.3.2 Upgrade Service

Upgrade or deploy a new configuration for the service. <service> defaults to all deployments in the namespace.

7.1.3.2.1 Syntax

```
mosadm upgrade [<service>]
```

7.1.3.2.2 Description

If you want to update the number of available MATLAB instances in the prewarmed MATLAB Pool:

1. In the MATLAB Pool override file (matlab-pool.yaml), change the replicaCount attribute value to the desired number, for example: - replicaCount: 4
2. Run: `mosadm upgrade matlab-pool`

If you are updating a property that is applicable to all services:

1. In the all-services override file (all.yaml), update the desired property.
2. Run: `mosadm upgrade`

   Performing these steps upgrades all services by default.

7.1.3.2.3 Example Output

```
ubuntu@ip-172-31-81-54:~/matlab_online_server$ sudo ./mosadm upgrade
matlab-pool

Will perform upgrade

Chart to upgrade in matlab-online-server/mathworks: matlab-pool

Is the information displayed above correct (y/n)? y

Found deployments for matlab-pool: default

Upgrading deployment mathworks-matlab-pool

helm --kube-context kubernetes-admin@matlab-online-server upgrade
mathworks-matlab-pool charts/matlab-pool-chart -f overrides/matlab-
online-server/mathworks/all.yaml -f overrides/matlab-online-
server/mathworks/matlab-pool.yaml

Release "mathworks-matlab-pool" has been upgraded.

LAST DEPLOYED: Fri Jan 24 21:06:50 2020

NAMESPACE: mathworks

STATUS: DEPLOYED

RESOURCES:

==> v1/ConfigMap

NAME                                  DATA  AGE

mathworks-matlab-pool                 7     36m

mathworks-matlab-pool-apparmor-profiles  1     36m

==> v1/DaemonSet

NAME                                DESIRED  CURRENT  READY  UP-TO-
DATE  AVAILABLE  NODE SELECTOR  AGE

mathworks-matlab-pool-apparmor-
loader  1        1        1      1           1          <none>
36m

==> v1/Deployment

NAME                              READY  UP-TO-DATE  AVAILABLE  AGE

mathworks-matlab-pool             2/4    4           2          36m
```

```
mathworks-matlab-pool-helpsearch  1/1    1           1          36m

mathworks-matlab-pool-ui          1/1    1           1          36m

==> v1/NetworkPolicy

NAME                                  AGE

mathworks-matlab-pool-matlab-pool     36m

mathworks-matlab-pool-matlab-pool-ui  36m

==> v1/Pod(related)

NAME                                               READY  STATUS
       RESTARTS  AGE

mathworks-matlab-pool-749657898b-
6hnz5             0/4    ContainerCreating  0         0s

mathworks-matlab-pool-749657898b-
7x84t             4/4    Running            0         36m

mathworks-matlab-pool-749657898b-
fnrw2             0/4    ContainerCreating  0         0s

mathworks-matlab-pool-749657898b-
vnbs9             4/4    Running            0         36m

mathworks-matlab-pool-apparmor-loader-
29lfb        1/1    Running            0         36m

mathworks-matlab-pool-helpsearch-8547f5bdf4-
58hx6  1/1    Running            0         36m

mathworks-matlab-pool-ui-6c999698d4-
5kln9           1/1    Running            0         36m

==> v1/ReplicaSet

NAME                              DESIRED  CURRENT  READY  AGE

mathworks-matlab-pool-terminated  0        0        0      36m

==> v1/Service

NAME                              TYPE       CLUSTER-IP      EXTERNAL-
IP  PORT(S)  AGE

mathworks-matlab-pool-
helpsearch  ClusterIP  10.110.210.140  <none>       80/TCP   36m

mathworks-matlab-pool-
ui          ClusterIP  10.98.150.35    <none>       80/TCP   36m

==> v1beta1/Ingress

NAME                              AGE

mathworks-matlab-pool-helpsearch  36m

mathworks-matlab-pool-ui          36m
```

```
Chart upgrade complete
```

7.1.3.3 Undeploy Service

Remove the service from the cluster.

7.1.3.3.1 Syntax

```
     mosadm undeploy [<service>]
```

To undeploy all services, namely authnz, core-ui, gateway, license, matlab-pool, namespace, nginx-ingress, and resource, use the following format:

```
 mosadm undeploy
```

To undeploy a specific service, use the following format:

```
mosadm undeploy authnz

mosadm undeploy matlab-pool
```

7.1.3.3.2 Example Output

```
ubuntu@ip-172-31-81-54:~/matlab_online_server$ sudo ./mosadm undeploy

Will perform undeploy

Charts to undeploy in matlab-online-server/mathworks:

authnz core-ui gateway license matlab-pool namespace nginx-ingress
resource

Is the information displayed above correct (y/n)? y

Found deployments for authnz: default

Deleting deployment mathworks-authnz

helm --kube-context kubernetes-admin@matlab-online-server delete --
purge mathworks-authnz

release "mathworks-authnz" deleted

Found deployments for core-ui: default

Deleting deployment mathworks-core-ui

helm --kube-context kubernetes-admin@matlab-online-server delete --
purge mathworks-core-ui

release "mathworks-core-ui" deleted
```

```
Found deployments for gateway: default

Deleting deployment mathworks-gateway

helm --kube-context kubernetes-admin@matlab-online-server delete --
purge mathworks-gateway

release "mathworks-gateway" deleted

Found deployments for license: default

Deleting deployment mathworks-license

helm --kube-context kubernetes-admin@matlab-online-server delete --
purge mathworks-license

release "mathworks-license" deleted

Found deployments for matlab-pool: default

Deleting deployment mathworks-matlab-pool

helm --kube-context kubernetes-admin@matlab-online-server delete --
purge mathworks-matlab-pool

release "mathworks-matlab-pool" deleted

Found deployments for namespace: default

Deleting deployment mathworks-namespace

helm --kube-context kubernetes-admin@matlab-online-server delete --
purge mathworks-namespace

release "mathworks-namespace" deleted

Found deployments for nginx-ingress: default

Deleting deployment mathworks-nginx-ingress

helm --kube-context kubernetes-admin@matlab-online-server delete --
purge mathworks-nginx-ingress

release "mathworks-nginx-ingress" deleted

Found deployments for resource: default

Deleting deployment mathworks-resource

helm --kube-context kubernetes-admin@matlab-online-server delete --
purge mathworks-resource

release "mathworks-resource" deleted

Chart undeploy complete
```

7.1.4 System Information

Gather system information, specifically Unix system information, the present working directory (tree), and disk usage, and display it to the terminal as well as log it to the install.log file.

7.1.4.1 Syntax

```
sudo ./mosadm gather-system-info
```

7.2 Bootstrap or Reset Single-Node Kubernetes Cluster

Use mosadm to bootstrap or to reset the cluster.

> **Note:** For RedHat/CentOS, set the SELINUX profile to permissive mode. The following commands can be executed with sudo privileges.
>
> ```
> # Set SELinux in permissive mode (effectively disabling it)
> 1. setenforce 0
> 2. sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/'
>    /etc/selinux/config
> ```

7.2.1 Bootstrap Node

Use the mosadm command to install Kubernetes and bootstrap the node as a single-node Kubernetes master.

7.2.1.1 Syntax

```
mosadm bootstrap-node
```

7.2.1.2 Description for Ubuntu Systems

When bootstrap-node is executed on an Ubuntu distribution, the following changes are applied to the machine.

After the operating system is configured with the following settings, the Kubernetes single node cluster is bootstrapped.

- Coredns is installed (to allow DNS lookups within the Kubernetes pods)
- Kube-proxy is removed
- Kube-router is installed as the container network interface and proxy server, and the network policy provider.
- The current node is untainted to allow Kubernetes to schedule pods.

7.2.1.2.1 Extra repositories

The following repositories are added to the package manager apart from the pre-built defaults:

- https://packages.cloud.google.com/apt/doc/apt-key.gpg
- http://apt.kubernetes.io kubernetes-xenial main

7.2.1.2.2 Updates

`apt-get update` is called within the script to update all the base utilities of the operating system.

7.2.1.2.3 Installed Utilities

The following operating system utilities are installed as part of the bootstrap process:

- unzip
- gettext
- jq
- ca-certificates
- curl
- software-properties-common
- git
- nfs-common
- ipvsadm
- kubelet (1.15.3)
- kubeadm (1.15.3)
- kubectl (1.15.3)
- helm (2.15.2)

7.2.1.2.4 Networking

The script allows and configures the following:

- Bridge networking
- IP forwarding
- Enables the following IPVS modules:
  - Ip_vs_dh
  - ip_vs_ftp
  - ip_vs
  - ip_vs_lblc
  - ip_vs_lblcr
  - ip_vs_lc
  - ip_vs_nq
  - ip_vs_rr
  - ip_vs_sed
  - ip_vs_sh
  - ip_vs_wlc
  - ip_vs_wrr
- Enables firewall
  - By default, only ssh (22), http (80), and https (443) are allowed.
  - All other ports are disabled if `ufw` is already installed (on Ubuntu distribution, `ufw` is installed by default).

7.2.1.2.5 Docker Daemon

Unless bootstrap-node is executed with the flag `–preserve-docker-config`, bootstrap node creates or overwrites the file `/etc/docker/daemon.json` with the following properties:

```
{
	"exec-opts": ["native.cgroupdriver=systemd"],
        "log-driver": "json-file",
        "log-opts": {
          "max-size": "100m"
        },
        "storage-driver": "overlay2"
}
```

Refer to the topic Daemon configuration file on Docker docs for more details.

7.2.1.2.6 Miscellaneous

- Enables high precision timestamps in syslog.
- Disables swap (disabling swap is required by the Kubernetes/kubelet to bootstrap).

7.2.1.3 Description for RedHat/CentOS Systems

When bootstrap-node is executed on RedHat/CentOS distributions, the following changes are applied to the machine.

After the operating system is configured with the following settings, Kubernetes single node cluster is bootstrapped.

- Coredns is installed, to allow DNS lookups within the Kubernetes pods.
- Kube-proxy is removed.
- Kube-router is installed as the container network interface and proxy server, and the network policy provider.
- The current node is untainted to allow Kubernetes to schedule pods.

7.2.1.3.1 Extra repositories

The following repositories are added to the package manager apart from the pre-built defaults:

- https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
    - https://packages.cloud.google.com/yum/doc/yum-key.gpg (key)
    - https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg (key)

7.2.1.3.2 Updates

`yum update` is called within the script to update all the base utilities of the operating system.

7.2.1.3.3 Installed Utilities

The following operating system utilities are installed as part of the bootstrap process:

- unzip
- gettext
- ca-certificates
- curl
- git
- nfs-utils
- ipvsadm

- yum-plugin-versionlock
- kubelet (1.15.3)
- kubeadm (1.15.3)
- kubectl (1.15.3)
- helm (2.15.2)

7.2.1.3.4 Networking

The script allows and configures the following:

- Bridge networking
- IP forwarding
- Enables the following IPVS modules:
    - Ip_vs_dh
    - ip_vs_ftp
    - ip_vs
    - ip_vs_lblc
    - ip_vs_lblcr
    - ip_vs_lc
    - ip_vs_nq
    - ip_vs_rr
    - ip_vs_sed
    - ip_vs_sh
    - ip_vs_wlc
    - ip_vs_wrr
    - br_netfilter
- Enables firewall
    - By default, only ssh (22), http (80), and https (443) are allowed.
    - All other ports are disabled if `firewalld` is already installed (on RedHat distribution, `firewalld` is installed by default).

7.2.1.3.5 Docker Daemon

Unless bootstrap-node is executed with the flag `–preserve-docker-config`, bootstrap node creates or overwrites the file `/etc/docker/daemon.json` with the following properties:

```
{
      "exec-opts": ["native.cgroupdriver=systemd"],
      "log-driver": "json-file",
      "log-opts": {
        "max-size": "100m"
      },
      "storage-driver": "overlay2"
}
```

Refer to the topic Daemon configuration file on docker docs for more details.

7.2.1.3.6 Miscellaneous

- The script enables high precision timestamps in syslog.
- Disables swap (disabling swap is required by the Kubernetes/kubelet to bootstrap)

7.2.2 Reset Node

Reset a Kubernetes cluster.

Use mosadm with the reset-node option to perform the following actions:

- Stop kubelet on the machine.
- Remove the Kubernetes cluster running locally.
- Reset the IP tables.

Run this command for a single-node cluster that was created using mosadm bootstrap-node.

After you run this command, you can re-run the bootstrap-node command to redeploy a single-node Kubernetes cluster to continue working with MATLAB Online Server. The reset-node command does not uninstall Linux packages deployed during the bootstrap-node script. To reset your environment fully, re-image the operating system or restore it from a backup.

7.2.2.1 Syntax

```
mosadm reset-node
```

7.2.2.2 Example Output

```
ubuntu@ip-172-31-67-165:~/matlab_online_server$ sudo ./mosadm reset-
node

Will reset the kubernetes node and attempt to clean up

Is the information displayed above correct (y/n)? y

./data/kubepacker-direct-install-1.0.1/files/opt/mw/uninstall.sh

[reset] Reading configuration from the cluster...

[reset] FYI: You can look at this config file with 'kubectl -n kube-
system get cm kubeadm-config -oyaml'

W0128 21:05:54.216468   32264 reset.go:98] [reset] Unable to fetch the
kubeadm-config ConfigMap from cluster: failed to get component
configs: configmaps "kube-proxy" not found

[reset] WARNING: Changes made to this host by 'kubeadm init' or
'kubeadm join' will be reverted.

[reset] Are you sure you want to proceed? [y/N]: y

[preflight] Running pre-flight checks
```

```
W0128 21:05:56.749305   32264 removeetcdmember.go:79] [reset] No
kubeadm config, using etcd pod spec to get data directory

[reset] Stopping the kubelet service

[reset] Unmounting mounted directories in "/var/lib/kubelet"

[reset] Deleting contents of config directories:
[/etc/kubernetes/manifests /etc/kubernetes/pki]

[reset] Deleting files: [/etc/kubernetes/admin.conf
/etc/kubernetes/kubelet.conf /etc/kubernetes/bootstrap-kubelet.conf
/etc/kubernetes/controller-manager.conf
/etc/kubernetes/scheduler.conf]

[reset] Deleting contents of stateful directories: [/var/lib/etcd
/var/lib/kubelet /etc/cni/net.d /var/lib/dockershim
/var/run/kubernetes]

The reset process does not reset or clean up iptables rules or IPVS
tables.

If you wish to reset iptables, you must do so manually.

For example:

iptables -F && iptables -t nat -F && iptables -t mangle -F && iptables
-X

If your cluster was setup to utilize IPVS, run ipvsadm --clear (or
similar)

to reset your system's IPVS tables.

The reset process does not clean your kubeconfig files and you must
remove them manually.

Please, check the contents of the $HOME/.kube/config file.
```

7.3 Configure Overrides

Configure access to Kubernetes clusters and use Helm charts and overrides to define and manage Kubernetes applications.

7.3.1 Kubernetes Configuration

7.3.1.1 Merge Kubernetes Config Files

Merge the cluster Kubernetes configuration (/etc/kubernetes/admin.conf) into the user Kubernetes configuration (~/.kube/config) to make it easier to run the kubectl command line tool.

7.3.1.1.1 Syntax

```
mosadm merge-kube-config
```

7.3.1.1.2 Example Output

```
Using kubeconfig from /etc/kubernetes/admin.conf

Will merge the current config from ~/.kube/config with the one in /etc/kubernetes/admin.conf

mkdir -p /home/ubuntu/.kube

cp /etc/kubernetes/admin.conf /home/ubuntu/.kube/config

chown -R ubuntu:ubuntu /home/ubuntu/.kube
```

7.3.2 Copy Helm Charts

Copy Helm charts into the install location.

7.3.2.1 Syntax

```
mosadm copy-helm-charts
```

7.3.2.2 Example Output

```
ubuntu@ip-172-31-67-165:~/matlab_online_server$ ./mosadm copy-helm-charts

Will copy helm charts to /home/ubuntu/matlab_online_server/charts

Is the information displayed above correct (y/n)? y

mkdir -p /home/ubuntu/matlab_online_server/charts

cp -rf ./data/com.mathworks.matlabonlineserver.matlab-pool-chart-1.0.5-SNAPSHOT /home/ubuntu/matlab_online_server/charts/matlab-pool-chart

…

Helm charts copied to /home/ubuntu/matlab_online_server/charts
```

7.3.3 Generate Overrides

Generate Helm override files based on install.config.

7.3.3.1 Syntax

```
mosadm generate-overrides
```

7.3.3.2 Example Output

If you are trying to generate overrides when the MATLAB install or MATLAB Docker image is not found, you might see the following output:

```
ubuntu@ip-172-31-67-165:~/matlab_online_server$ sudo ./mosadm
generate-overrides

Will generate overrides in
/home/ubuntu/matlab_online_server/overrides

Is the information displayed above correct (y/n)? y

Did not find a MATLAB docker image, using MATLAB from the host
(/usr/local/MATLAB/R2020a)

Did not find an installation of MATLAB at the expected location:
/usr/local/MATLAB/R2020a
```

Set up the MATLAB install or MATLAB docker image, and then try the command again. The output you see resembles the following:

```
ubuntu@ip-172-31-67-165:~/matlab_online_server$ ./mosadm generate-
overrides

Will generate overrides in
/home/ubuntu/matlab_online_server/overrides

Is the information displayed above correct (y/n)? y

Found MATLAB image in the docker registry, using it instead of host
MATLAB

mathworks/mos/com.mathworks.matlabonlineserver.matlab-image:R2020a

chmod u+w ./install.config

mkdir -p /home/ubuntu/matlab_online_server/overrides/matlab-online-
server/mathworks

Generating overrides in
/home/ubuntu/matlab_online_server/overrides/matlab-online-
server/mathworks

…

Generating override for authnz
Generating override for matlab-pool
```

7.3.4 Additional Configuration Options

Description	Syntax
Force acceptance of terms of use and do not show a prompt before running the operation.	--skip-prompt | -y
Override the cluster to be used for the installation.	--cluster *cluster-name*
Set the namespace to be used for the installation.	--namespace *your-namespace*
Override the location of the installation data folder.	--data-location */path/to/data*
Set the chart directory.	--charts-dir */path/to/charts*
Configure the location of overrides.	--overrides-dir */path/to/overrides*
Path to the installation configuration file.	--install-config */path/to/install.config*
Path to the Kubernetes configuration file.	--kube-config /path/to/kubeconfig
Path to the file on disk for storing the log.	--log-file /path/to/logfile

8 Monitoring

8.1 Check Pod Status

Check the status of the Kubernetes cluster pods. Before you can check the pods, you must have `kubectl` installed and configured to access the MATLAB Online Server Kubernetes cluster and have `helm` installed on the same nodes as `kubectl`.

8.1.1 Get Pod List

MATLAB Online Server microservices run on Kubernetes cluster pods. To display a list of these pods, run the `kubectl` command with `get pods`.

8.1.1.1 Syntax

Issue the following command, replacing the variable *your-namespace* with the namespace you used to deploy MATLAB Online Server:

```
$ kubectl --namespace=your-namespace get pods
```

8.1.1.2 Example Output

This command results in a display of running pods and statuses:

```
                    NAME                                      READY   STATUS    RESTARTS   AGE
your-namespace-authnz-authnz-5bd9867d69-2trk2                  1/1    Running      0       17h
your-namespace-core-ui-core-ui-6679d5ff7f-rdtfb                1/1    Running      0       17h
your-namespace-gateway-65697586f7-vsvls                        1/1    Running      0       17h
your-namespace-license-84d4877898-mmqll                        1/1    Running      0       17h
your-namespace-matlab-pool-5858d796b9-lqslb                    4/4    Running      0       14m
your-namespace-matlab-pool-apparmor-loader-6prmt               1/1    Running      0       17h
your-namespace-matlab-pool-apparmor-loader-99zcq               1/1    Running      0       17h
your-namespace-matlab-pool-helpsearch-5689dd4cdf-
t77rh                                                          1/1    Running      0       17h
your-namespace-matlab-pool-ui-689fb5d9ff-tsrh4                 1/1    Running      0       17h
your-namespace-resource-resource-56f5745747-dctps              1/1    Running      0       17h
```

The information returned can be broken down as follows:

- The following services are MATLAB Online Server core services:
  - authnz
  - core-ui
  - gateway
  - license
  - helpsearch
  - ui
  - resource

  All these services are expected to have the value 1/1 under the READY column.
- The MATLAB Pool pod hosts the MATLAB process and it is expected to have the value 4/4 under the READY column. MATLAB Online Server assigns a separate pod for each MATLAB Online Server user.

- matlab-pool-apparmor-loader is a specific pod that MATLAB Online Server deploys for each Kubernetes node. The display shown in the previous example lists two pods of this type because its Kubernetes cluster has two nodes.

The name of each pod has suffix at the end as a unique identifier. As an example, the authnz-authnz pod name ends with the suffix 5bd9867d69-2trk2. If a new pod is created, then you get a new suffix identifier at the end of its name.

8.1.2 Get List of Other Required Services

While the command `kubectl` lists the main MATLAB Online Server services, running that command does not make visible other required services, such as namespace. Instead, check the deployment status of other required services by running the `helm list` command.

8.1.2.1 Syntax

In this example, replace *your-namespace* with the namespace you used for MATLAB Online Server:

```
helm list | egrep "your-namespace|nginx"
```

8.1.2.2 Example Output

This command results in a display of both core and additional services and their status.

```
                          Thu Jan 16
your-namespace-           16:51:39
authnz                 1  2020        DEPLOYED  authnz-1.0.0     1.0.0    your-namespace

                          Thu Jan 16
your-namespace-core-      16:51:40
ui                     1  2020        DEPLOYED  core-ui-1.0.0    1.0.0     your-namespace

                          Thu Jan 16
your-namespace-           16:51:41
gateway                1  2020        DEPLOYED  gateway-1.0.0    1.0.0    your-namespace

                          Thu Jan 16
your-namespace-           16:51:42
license                1  2020        DEPLOYED  license-1.0.0    1.0.0    your-namespace

                          Thu Jan 16
your-namespace-           16:51:43              matlab-pool-
matlab-pool            1  2020        DEPLOYED  1.0.0            1.0.0    your-namespace

                          Thu Jan 16
your-namespace-           16:51:45
namespace              1  2020        DEPLOYED  namespace-1.0.0  1.0.0    your-namespace
```

```
                           Thu Jan 16
your-namespace-            16:51:46
resource                1  2020        DEPLOYED  resource-1.0.0   1.0.0    your-namespace

nginx-ingress           3  Sat Nov 23  DEPLOYED  nginx-ingress-   0.26.1c  nginx-ingress
                           15:21:31              1.25.0
                           2019
```

Make sure each service has the value DEPLOYED.

8.1.3 Get Pod Descriptions

To get more details about each pod, run the following command:

```
kubectl describe pod podId
```

8.1.3.1 Syntax

For example, issue the following command, replacing *your-namespace* with the namespace you used for MATLAB Online Server:

```
kubectl --namespace=your-namespace describe pod your-namespace-matlab-pool-5858d796b9-lqslb
```

This command provides details about the MATLAB Pool pod identified with `your-namespace-matlab-pool-5858d796b9-lqslb`.

8.1.3.2 Example Output

Pod Descriptors	
Name	*your-namespace*-matlab-pool-5858d796b9-lqslb
Namespace	*your-namespace*
Node	private-node-3/123.45.67.89
Start Time	Fri, 17 Jan 2020 10:25:20 -0500
Status	Running
IP	987.65.43.21
Containers	matlab: Container ID: docker://cf5d62aeb248a87ddac5fdd775cae7c94e8df06816ad68c5d1db78b87f903cb4
Port	8080/TCP

Pod Descriptors	
Events	Type Reason Age From Message ------ ---------- ------ --- ---- ------------ **Normal** Scheduled <unknown> default-scheduler Successfully assigned <*your-namespace*>/<*your-namespace*>-matlab- pool-5858d796b9-lqslb to private-node-3 **Normal** Pulling 19m kubelet, private-node-3 Pulling image " mathworks.com/matlabonlineserver/com.mathworks.matlabonlineserver. matlab-omnibus-debian-image:R2020a" **Normal** Pulled 19m kubelet, private-node-3 Successfully pulled image " mathworks.com/matlabonlineserver/com.mathworks.matlabonlineserver. matlab-omnibus-debian-image:R2020a" **Normal** Created 19m kubelet, private-node-3 Created container matlab **Normal** Pulling 19m kubelet, private-node-3 Pulling image " mathworks.com/matlabonlineserver/com.mathworks.matlabonlineserver. matlab-omnibus-debian-image:R2020a"

Pod Descriptors	
Events (continued)	**Normal** Pulling 19m kubelet, private-node-3 Pulling image " mathworks.com/matlabonlineserver/com.mathworks.matlabonlineserver. omnibus-debian-image:R2020a" **Normal** Pulled 19m kubelet, private-node-3 Successfully pulled image " mathworks.com/matlabonlineserver/com.mathworks.matlabonlineserver. omnibus-debian-image:R2020a" **Normal** Created 19m kubelet, private-node-3 Created container resource-proxy **Normal** Started 19m kubelet, private-node-3 Started container resource-proxy **Normal** Started 19m kubelet, private-node-3 Started container matlab **Normal** Pulled 19m kubelet, private-node-3 Successfully pulled image " mathworks.com/matlabonlineserver/com.mathworks.matlabonlineserver. matlab-omnibus-debian-image:R2020a" **Normal** Created 19m kubelet, private-node-3 Created container display **Normal** Started 19m kubelet, private-node-3 Started container display **Normal** Pulling 19m kubelet, private-node-3 Pulling image " mathworks.com/matlabonlineserver/com.mathworks.matlabonlineserver. matlab-omnibus-debian-image:R2020a" **Normal** Pulled 19m kubelet, private-node-3 Successfully pulled image " mathworks.com/matlabonlineserver/com.mathworks.matlabonlineserver. matlab-omnibus-debian-image:R2020a" **Normal** Created 19m kubelet, private-node-3 Created container windowmanager **Normal** Started 19m kubelet, private-node-3 Started container windowmanager

8.2 Check Logs

You can get the content of MATLAB Online Server microservice logs in one of several ways. Choose the method that works best for you.

8.2.1 Access Logs Using the File System

Access the single node on which you installed MATLAB Online Server (for example, using ssh), and then access the folder /var/log/containers/. This folder contains logs for all the MATLAB Online Server microservices.

Issue the following command, replacing *your-namespace* with the namespace you used for MATLAB Online Server:

```
ls /var/log/containers/your-namespace
```

8.2.1.1 Examples:

your-namespace-**core-ui**-core-ui-6679d5ff7f-rdtfb_ *your-namespace* _core-ui-165cfda8ff5c4d2b01838d0acb573df881e04a08c329afcda1b96b9c15cd0450.log

your-namespace -gateway-65697586f7-vsvls_ *your-namespace* _gateway-78afff6773618381f1fc6ba05e63eceaf3f3a6fd5de3defd6f50b28a97bb7a44.log

your-namespace -matlab-pool-apparmor-loader-zdqwk_ *your-namespace* _apparmor-loader-87115017a99407c3bf647712c61426d9f226a8b4bb09c33ba915761bd61a043d.log

your-namespace -matlab-pool-helpsearch-5689dd4cdf-t77rh_ *your-namespace* _helpsearch-963806b3d47815d8c7df217770b38be10c37fc30b5f3c40ba7c2cc4084dc4ce0.log

your-namespace -matlab-pool-ui-689fb5d9ff-tsrh4_ *your-namespace* _webgui-ec90a23bbd0db048115f6fae4113084a8df6d0ef860fc03dc54e9ffb49d87a3a.log

8.2.2 Dump Logs to System Output

Use the kubectl command to access the logs for individual pods. The following example demonstrates the command to dump the content of the logs belonging to the matlab-pool-helpsearch pod. Replace *your-namespace* with the namespace you used for MATLAB Online Server.

```
kubectl --namespace= your-namespace logs your-namespace -matlab-pool-helpsearch-5689dd4cdf-t77rh
```

If a pod has more than one container, for example, matlab-pool, you can first extract the list of its containers, and then dump the content of its container logs by specifying the name of the container.

8.2.2.1 Examples

To get the list of containers for the MATLAB Pool pod, enter the following command, replacing *your-namespace* with the namespace you used for MATLAB Online Server:

```
kubectl --namespace= your-namespace get pods your-namespace -matlab-pool-5858d796b9-lqslb -o jsonpath='{.spec.containers[*].name}'
```

The results list four containers inside the MATLAB Pool pod. The containers are listed left to right with a space between each container name:

matlab resource-proxy display windowmanager

To get logs for the matlab container that is running inside the MATLAB Pool pod, run the following command, replacing *your-namespace* with the namespace you used for MATLAB Online Server:

kubectl --namespace= *your-namespace* logs *your-namespace* -matlab-pool-5858d796b9-lqslb -c matlab

8.2.3 Stream Logs to System Output

To continuously monitor the contents of logs, stream their contents by adding the -f flag to the kubectl command as follows, replacing *your-namespace* with the namespace you used for MATLAB Online Server:

```
kubectl --namespace= your-namespace logs -f your-namespace -matlab-pool-helpsearch-
5689dd4cdf-t77rh
```

```
kubectl --namespace= your-namespace logs -f your-namespace -matlab-pool-5858d796b9-
lqslb -c matlab
```

8.3 Kubernetes Dashboard for MATLAB Online Server

Set up Kubernetes Dashboard (also known as Web UI) on a MATLAB Online Server single-node configuration. The example in this section can be the basis for a larger deployment and set of security configurations in your environment.

Before you begin, Kubernetes kubectl must be configured. See Step 5. Install Kubernetes in the MATLAB Online Server installation instructions.

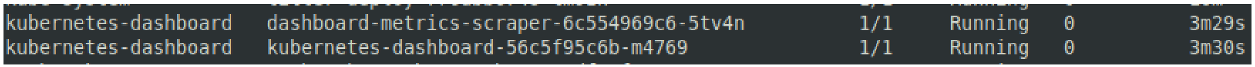
8.3.1 Install Kubernetes Dashboard

Before you begin, review the Kubernetes documentation at https://kubernetes.io/docs/tasks/access-application-cluster/web-ui-dashboard/, which provides detailed instructions and security configuration options.

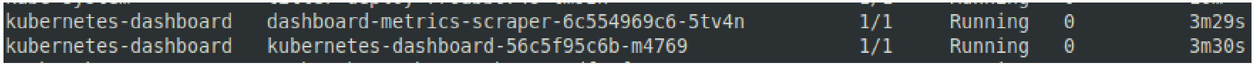
1. Install and deploy the dashboard by running the following command:

   ```
   kubectl apply -f
   https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0-
   beta8/aio/deploy/recommended.yaml
   ```

   If installation and deployment are successful, the terminal displays a new namespace with two pods:



```
kubernetes-dashboard   dashboard-metrics-scraper-6c554969c6-5tv4n      1/1     Running   0          3m29s
kubernetes-dashboard   kubernetes-dashboard-56c5f95c6b-m4769           1/1     Running   0          3m30s
```



```
kubernetes-dashboard   dashboard-metrics-scraper-6c554969c6-5tv4n      1/1     Running   0          3m29s
kubernetes-dashboard   kubernetes-dashboard-56c5f95c6b-m4769           1/1     Running   0          3m30s
```

2. Create a Kubernetes service account. Your account allows authentication into the dashboard. To create the account:
   a. Create the file `matlab-online-server-admin-account.yaml`.
   b. Enter the following information into the file (enter the information exactly as shown):

   ```
   apiVersion: v1
   kind: ServiceAccount
   metadata:
     name: matlab-online-server-admin
     namespace: kube-system
   ---
   apiVersion: rbac.authorization.k8s.io/v1beta1
   kind: ClusterRoleBinding
   metadata:
     name: matlab-online-server-admin
   roleRef:
     apiGroup: rbac.authorization.k8s.io
     kind: ClusterRole
     name: cluster-admin
   subjects:
     - kind: ServiceAccount
       name: matlab-online-server-admin
       namespace: kube-system
   ```

3. Apply the file to your Kubernetes cluster by issuing the following command:

```
kubectl apply -f matlab-online-server-admin-account.yaml
```

4. For local access to the Kubernetes Dashboard, retrieve the bearer token for the created matlab-online-server-admin account. To retrieve the bearer token string, issue the following command.

```
kubectl -n kube-system describe secret $(kubectl -n kube-system get
secret | grep matlab-online-server-admin | awk '{print $1}') | grep
token: | awk '{print $2}'
```

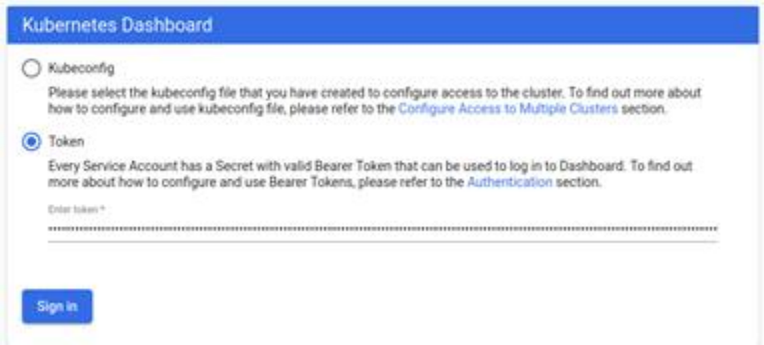
5. Run the following command to start a proxy to allow access to the Kubernetes Dashboard:

```
kubectl proxy
```

6. Access the Kubernetes Dashboard on a local browser at the following URL:

http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/login

7. Select 'Token' for the authentication method and then provide the token retrieved in a previous step:



Kubernetes Dashboard

○ Kubeconfig

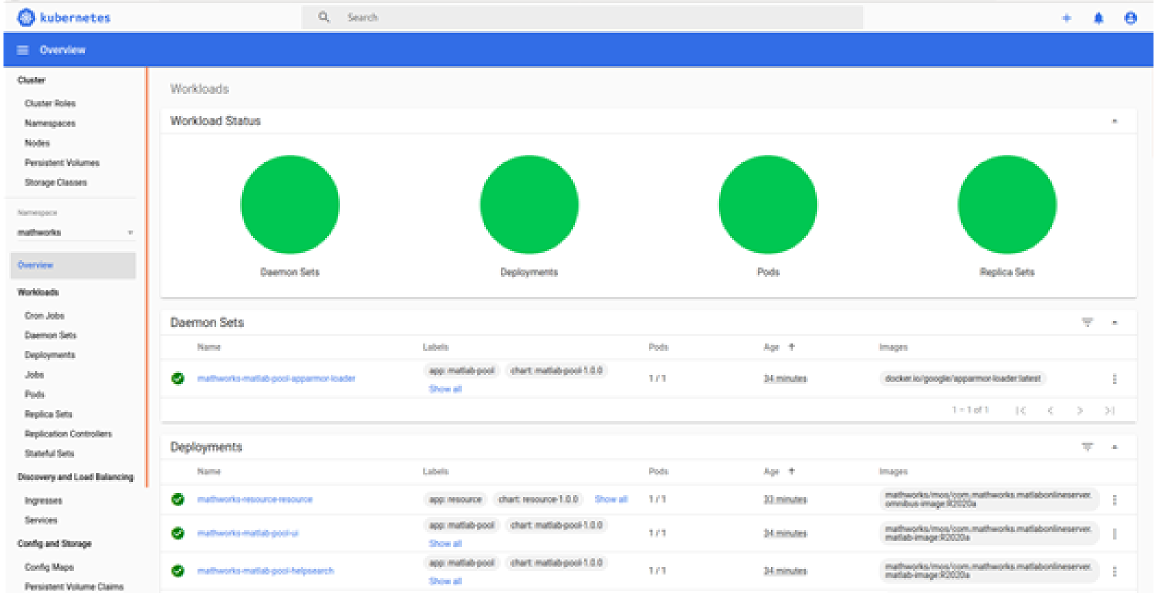
Please select the kubeconfig file that you have created to configure access to the cluster. To find out more about how to configure and use kubeconfig file, please refer to the Configure Access to Multiple Clusters section.

● Token

Every Service Account has a Secret with valid Bearer Token that can be used to log in to Dashboard. To find out more about how to configure and use Bearer Tokens, please refer to the Authentication section.

Enter token *

8. After you log in, familiarize yourself with the different namespaces for your environment and the administrative features in the Kubernetes Dashboard.



Daemon Sets

	Name	Labels	Pods	Age ↑	Images
✓	mathworks-matlab-pool-apparmor-loader	app: matlab-pool chart: matlab-pool-1.0.0 Show all	1 / 1	34 minutes	docker.io/google/apparmor-loader:latest

1 – 1 of 1

Deployments

	Name	Labels	Pods	Age ↑	Images
✓	mathworks-resource-resource	app: resource chart: resource-1.0.0 Show all	1 / 1	33 minutes	mathworks/mos/com.mathworks.matlabonlineserver.omnibus-image:R2020a
✓	mathworks-matlab-pool-ui	app: matlab-pool chart: matlab-pool-1.0.0 Show all	1 / 1	34 minutes	mathworks/mos/com.mathworks.matlabonlineserver.matlab-image:R2020a
✓	mathworks-matlab-pool-helpsearch	app: matlab-pool chart: matlab-pool-1.0.0 Show all	1 / 1	34 minutes	mathworks/mos/com.mathworks.matlabonlineserver.matlab-image:R2020a

8.3.2 Managed Kubernetes Providers

8.3.2.1 Kubernetes Dashboard with AWS

To set up Kubernetes Dashboard with Amazon Web Services managed Elastic Kubernetes Service, follow Amazon’s tutorial:

https://docs.aws.amazon.com/eks/latest/userguide/dashboard-tutorial.html

8.3.2.2 Kubernetes Dashboard with Azure

To set up Kubernetes Dashboard with Microsoft® Azure’s managed Azure Kubernetes Service, follow Microsoft’s tutorial:

https://docs.microsoft.com/en-us/azure/aks/kubernetes-dashboard

8.3.2.3 Kubernetes Dashboard with Google Kubernetes Engine

Google is deprecating the open source Kubernetes Dashboard on their managed Google Kubernetes Engine and recommends that you use their Cloud Console user interface.

8.4 Log Shipping

MATLAB Online Server does not come with any log shipping capabilities. If you are interested in persisting the MATLAB Online Server application logs for debugging and future use or in integrating with Splunk, follow the procedures in Check Logs.

8.4.1 Option 1: Persistence of Logs on Host

MATLAB Online Server comprises various services. Logs from multiple services can be fetched using kubectl and stored in files on disk.

For logs of MATLAB pods that are in use, perform the following steps:

1. Run the following command to get the status of in-use pods:

   ```
   kubectl get pods -l inUse=true
   ```

   This action returns the status of pods in use:

   ```
   ubuntu@ip-172-31-23-150:~$ kubectl get pods -n mathworks -l inUse=true
   NAME                                    READY   STATUS    RESTARTS   AGE
   your-namespace-matlab-pool-f479b5b96-d6kfk   4/4     Running   0          90m
   ```

2. To retrieve the logs of a particular pod, run the following commands, replacing *your-namespace* with the namespace you used for MATLAB Online Server, depending on which logs you want to retrieve:

   - License service logs

kubectl logs <license_pod_name> -n *your-namespace* >> <path_to_license_log_file_on_host>

   - AuthNZ service logs

kubectl logs <authnz_pod_name> -n *your-namespace* >> <path_to_authnz_log_file_on_host>

   - MATLAB logs

kubectl logs <matlab_pod_name> -n *your-namespace* **-c matlab** >> <path_to_matlab_log_file_on_host>

   - Resource proxy logs

kubectl logs <matlab_pod_name> -n *your-namespace* **-c resource-proxy** >> <path_to_resource_proxy_log_file_on_host>

These examples illustrate storing log files on the host. You can persist logs on any other location, for example, an NFS server of your organization, where they can be exported with logging tools of your choice for more analysis and for setting up alerts.

8.4.2 **Option 2: Integrate with Splunk (Advanced)**

From Splunk Connect for Kubernetes on EKS!: "Splunk Connect for Kubernetes is a collection of Helm charts that will deploy a Splunk-supported deployment of Fluentd to your Kubernetes cluster, complete with a Splunk-built Fluentd HEC plugin to ship logs and metadata, and a metrics deployment that captures your cluster metrics into Splunk's Metric Store for use with our new analysis workspace."

> **Note:** Depending on the platform you have chosen, you might have to make changes for Splunk integration to work. For resources, check the documentation from your product vendor.

8.4.2.1 Before You Begin

Before you integrate with Splunk, perform the following tasks.

1. Create a minimum of two Splunk indices:
   - One events index, which handles logs and objects (you can also create two separate indices for logs and objects)
   - One metrics index

   If you do not configure these indices, Splunk Connect for Kubernetes uses the defaults created in your HEC (HTTP Event Collector) token.

2. Create an HEC token if you do not already have one.

8.4.2.2 Configure and Install Splunk

1. Create a namespace for Splunk Connect for Kubernetes with the following command:

```
kubectl create ns mos-splunk-connect-k8s
```

2. Create the file `values.yaml` with the following contents (type lines exactly as shown):

```
kubernetes:
 clusterName: <cluster-name>
logLevel: warn
global:
 splunk:
  hec:
   token: <hec-token>
   host: <splunk-host>
   port: <splunk-port>
   protocol: http
   insecureSSL: true
   indexName: <splunk-index-name>
journalLogPath: /var/log/journal
logs:
 matlab-pool:
  from:
   pod: "*-matlab-pool"
   container: matlab
  multiline:
   firstline: /^\[#\|\d{4}-\d{1,2}/
  bootstrap:
   from:
    journald:
     unit: kubeadm.service
   timestampExtraction:
    regexp: \w(?<time>[0-1]\d[0-3]\d [^\s]*)
    format: "%m%d %H:%M:%S.%N"
   sourcetype: kube:kubeadm
  resource-proxy:
  from:
   pod: "*-matlab-pool"
   container: resource-proxy
  multiline:
   firstline: /^\[#\|\d{4}-\d{1,2}/
  bootstrap:
   from:
    journald:
     unit: kubeadm.service
   timestampExtraction:
    regexp: \w(?<time>[0-1]\d[0-3]\d [^\s]*)
    format: "%m%d %H:%M:%S.%N"
   sourcetype: kube:kubeadm
```

3. Install Splunk Connect for Kubernetes.

   ```
   helm install --name mos-splunk-connect --namespace mos-splunk-connect-k8s -f values.yaml https://github.com/splunk/splunk-connect-for-kubernetes/releases/download/1.3.0/splunk-connect-for-kubernetes-1.3.0.tgz
   ```

> **Note:** This procedure is just one method to integrate Splunk with Kubernetes. There are other methods that you can choose based on your requirements.
>
> There are also various log shipping options with Kubernetes that you can choose to run, depending on your organization's needs and availability of tools.

8.5 Problem Solving in MATLAB Online Server

When MATLAB Online Server is not working as expected, consult this topic to learn more about the possible reasons and what actions you can take to resolve the problem.

8.5.1 MATLAB Pod Issues

The MATLAB pod comprises four Docker containers, one of which is MATLAB Runtime.

When the services are running successfully and have no issues, then the status of pods is "Running."

However, if the services are having issues, pods can still show the status Running. This can be because, for example, the MATLAB Docker image was built incorrectly, or there are issues connecting with the FLEXlm server.

8.5.1.1 Syntax

Run the following command to display pod status, replacing *your-namespace* with the namespace you used for MATLAB Online Server:

```
kubectl get pods –n your-namespace
```

8.5.1.2 Example Output

This command outputs information similar to that shown in the following example:

```
ubuntu@ubuntu:~/matlab_online_server$ kubectl get pods --namespace your-namespace

NAME                                                        READY   STATUS     RESTARTS
AGE

your-namespace-core-ui-core-ui-5bd9b7c88b-fqbl4
      1/1     Running        0          111s

your-namespace-authnz-authnz-7c7999589c-
x74d9       1/1     Running        0          111s

your-namespace-gateway-669dd7865f-
t6l74             1/1     Running        0          110s

your-namespace-license-548f7b6655-
8skdt             1/1     Running        0          108s

your-namespace-matlab-pool-8c679f76-
c4h4n           4/4     Running        0          107s

your-namespace-matlab-pool-8c679f76-
cckkp           4/4     Running        0          107s

your-namespace-matlab-pool-ui-579fc8ff49-
2bz25      1/1     Running        0          107s

your-namespace-nginx-ingress-controller-
kcwwd       1/1     Running        0          104s

your-namespace-nginx-ingress-default-backend-f78c5f7b8-dvxm8

1/1     Running        0          104s

your-namespace-resource-resource-894447df7-
r9ngb    1/1     Running        0          103s
```

8.5.1.3 Problems and Resolutions

Check the possible problems and solutions in this section to help diagnose and resolve common issues.

8.5.1.3.1 License Checkout Issues

Although a license checkout error is a failing scenario, the MATLAB pod status still shows as Running because a prewarmed pool of MATLAB instances might run successfully but fail to check out a license when a user signs in.

To determine if a license checkout error is the issue, perform the following assessment:

1. Run the following command to get the MATLAB container logs, replacing *matlab_pod_name* with the name of your MATLAB Online Server MATLAB pod and *your-namespace* with the namespace you used for MATLAB Online Server:

```
kubectl logs matlab_pod_name -n your-namespace -c matlab
```

2. Check for License Manager Error -<n>: License Checkout Failed in the logs. If this error has occurred, then it is highly likely that MATLAB is having issues checking out the license.

Logs can be lengthy. You can use a text-searching tool to get the information you need faster. For example, you can search the logs for a license checkout error:

```
kubectl logs matlab_pod_name -n your-namespace -c matlab | grep 'License Checkout'
```

You can also search for a license manager error:

```
kubectl logs matlab_pod_name -n your-namespace -c matlab | grep 'License Manager Error'
```

If you see License Manager Error -<n>: License Checkout Failed in the logs, check the specific error message, and then consult the following table.

Error Message	Resolution Steps
Invalid license file syntax	Check to make sure that the license file syntax is appropriate (that is, check the FLEXlm logs). Then, making sure that you use the correct license file, try again.

<table>
<tr><th>Error Message</th><th>Resolution Steps</th></tr>
<tr><td>Cannot connect to license server</td><td>

- Try to connect to the license server locally.
- Check to see if the network firewall rules for the license server are set correctly to allow connection from the MATLAB Online Server cluster.
- Network policy on a MATLAB pod is enabled by default, which allows it to talk to FLEXlm only on ports 27000 and 27001 by default (controlled by license.dat). Make sure your FLEXlm server is listening on those ports. If your server is listening on other ports, you need to change the following configuration in the matlab-pool override and restart the pod:

```
networkPolicy:
  enabled: true
  egress:
      ports:
      - port: <flexlm_port_here>
          protocol:
```

</td></tr>
</table>

8.5.1.3.2 Incorrect MATLAB Image

The MATLAB Image is built from a local MATLAB install and can sometime have issues if it was built from an incorrect MATLAB location or version that is not supported by MATLAB Online Server.

To determine if an incorrect MATLAB image is the problem, perform the following assessment.

1. Run the following command to display a pod status list, replacing *your-namespace* with the namespace you used for MATLAB Online Server:

```
kubectl get pods –n your-namespace
```

2. Examine the pod status information, as shown in this example:

```
ubuntu@ip-172-31-23-150:~/matlab_online_server$ kubectl get pods -n your-namespace

NAME                                     READY   STATUS    RESTARTS   AGE

your-namespace-authnz-authnz-569df55f78-pqklk      1/1     Running   0        47h

your-namespace-core-ui-core-ui-84c59ddd45-fwf4t    1/1     Running   0        47h

your-namespace-gateway-84dbf7c8d-6gjcz           1/1     Running   0        47h

your-namespace-license-85ffb6c595-q6x4f          1/1     Running   0        47h

your-namespace-matlab-pool-apparmor-loader-jzqsg   1/1     Running   0        40s

your-namespace-matlab-pool-f479b5b96-59xmj       3/4     Running   0        40s
```

```
your-namespace-matlab-pool-helpsearch-546cb5fd8-jkx4p

1/1     Running   0        40s

your-namespace-matlab-pool-ui-7b6d688f98-rfdz5       1/1     Running   0        40s

your-namespace-nginx-ingress-controller-5c6b5f775f-7ls87

1/1     Running   0        27h

your-namespace-nginx-ingress-default-backend-f78c5f7b8-8q4sj

1/1     Running   0        47h

your-namespace-resource-resource-8554f87bbf-g4ssq   1/1     Running   0        47h
```

3. If you see MATLAB pods in a 3/4 Running state, as shown in the example log, run the following command to learn why. Replace *matlab_pod_name* with your MATLAB Online Server MATLAB pod name and *your-namespace* with the namespace you used for MATLAB Online Server:

```
kubectl logs matlab_pod_name -n namespace -c matlab
```

This command outputs a log similar to this example:

```
Starting MATLAB

+ exec /MATLAB/bin/matlab -prewarm -c '"27000@flexlm.mwcloudtest.com"' -nodesktop -
softwareopengl -nosplash

MATLAB is selecting SOFTWARE OPENGL rendering.

00000003 2020-Feb-12 19:46:27.286432 0x00007f2b5858f8c0 fl::log I Now logging to
console stream

00000004 2020-Feb-12 19:46:27.286544 0x00007f2b5858f8c0 fl::log I Diagnostic logging
enabled & configured


< M A T L A B (R) >

Copyright 1984-2019 The MathWorks, Inc.

R2019b Update 5 (9.7.0.1310180) 64-bit (glnxa64)

February 7, 2020


Warning: Unrecognized command line option: prewarm.


To get started, type doc.

For product information, visit www.mathworks.com.
```

The logs show that MATLAB was not able to start successfully. To verify that this is what happened, you can run the following command to look at the logs of resource-proxy, which is another container in the MATLAB pod that proxies requests to MATLAB.

Issue this command, replacing *matlab_pod_name* with your MATLAB Online Server MATLAB pod name and *your-namespace* with the namespace you used for MATLAB Online Server:

```
kubectl logs matlab_pod_name -n your-namespace -c resource-proxy
```

Examine the logs output by this command. If you see the following messages, the backend (in this case, MATLAB) is not ready and is not functioning properly.

```
2020-02-12 19:06:51,037 INFO matlabonlineserver.resource_proxy Backend is not ready

2020-02-12 19:06:51,507 INFO matlabonlineserver.resource_proxy Backend is not ready

2020-02-12 19:06:54,498 WARNING matlabonlineserver.http.client Jetty request failed:
Connection refused

2020-02-12 19:07:04,499 WARNING matlabonlineserver.http.client Jetty request failed:
Connection refused

2020-02-12 19:07:11,037 INFO matlabonlineserver.resource_proxy Backend is not ready

2020-02-12 19:07:11,507 INFO matlabonlineserver.resource_proxy Backend is not ready

2020-02-12 19:07:14,500 WARNING matlabonlineserver.http.client Jetty request failed:
Connection refused

2020-02-12 19:07:24,500 WARNING matlabonlineserver.http.client Jetty request failed:
Connection refused

2020-02-12 19:07:31,037 INFO matlabonlineserver.resource_proxy Backend is not ready

2020-02-12 19:07:31,507 INFO matlabonlineserver.resource_proxy Backend is not ready
```

In the example log, the message "Backend is not ready" indicates that that MATLAB is not yet ready.

For resolution, check to see if you are building the MATLAB image from the correct MATLAB install location, which is R2020a only.

8.5.1.3.3 Storage Mount Errors

When all services and MATLAB are in a Running state, but you are encountering issues using MATLAB Online, it is possible that there was a problem with mounting the storage on MATLAB at the time of user sign on.

To determine if storage mounting on MATLAB is the problem, perform the following assessment:

1. Run the following command to generate pod logs, replacing *matlab_pod_name* with the name of your MATLAB Online Server MATLAB pod and *your-namespace* with the namespace you used for MATLAB Online Server.

   ```
   kubectl logs matlab_pod_name -n your-namespace -c resource-proxy
   ```

2. Examine the output from this command:

   ```
   2020-02-12 20:44:39,124 INFO matlabonlineserver.storage Setting up the
   profile: nfsHome
   ```

```
2020-02-12 20:44:39,124 INFO matlabonlineserver.storage Setting up the
mount: home

2020-02-12 20:44:39,124 INFO matlabonlineserver.storage Starting NFS mount

2020-02-12 20:44:39,125 INFO matlabonlineserver.storage Executing: mount -o
rw,relatime,vers=3,rsize=1048576,wsize=1048576,namlen=255,acregmin=600,acre
gmax=600,acdirmin=600,acdirmax=600,hard,nocto,noacl,proto=tcp,timeo=600,ret
rans=2,mountproto=tcp,local_lock=none,nolock nfs:/exports/home/oin
/mounts/homeoin

2020-02-12 20:44:49,167 INFO matlabonlineserver.storage mount exit value=
137, mount result= , mount error=
```

3. If you see the mount error message, as displayed in the last line of the example shown, check that the ports allowed by the network policy for MATLAB Pool include ports for your NFS server setup. For more details on network policies, see Configure Network Policies.

8.5.1.3.4 MATLAB Pod Not in Running State

If none of the possible issues described in this section help you to resolve the problem, try the following:

- See MATLAB Pod Issues.
- Contact MathWorks Support.
  1. Run the following `kubectl` commands, replacing *license_pod_name* with the name of the pod that is stuck and *your-namespace* with the namespace you used for MATLAB Online Server:

     ```
     kubectl describe pod license-pod-name --namespace your-namespace

     kubectl logs license-pod-name --namespace your-namespace
     ```

  2. Keep the output nearby to discuss with a support technician.

8.5.2 **Pod Creation Stuck**

During MATLAB Online Server installation and configuration, pods can sometimes get stuck in one of the following states:

- ImagePullBackOff
- CrashLoopBackOff
- Pending
- ContainerCreating

To see pod status, run the following command, replacing *your-namespace* with the namespace you used for MATLAB Online Server:

kubectl get pods –n *your-namespace*

Consult the following sections for instructions on correcting the pod state when the pod is stuck during creation.

8.5.2.1 ImagePullBackOff

To find out the exact error message for why the pod status is in ImagePullBackOff status, run the following command, replacing *pod_name* with the name of the pod that is stuck and *your-namespace* with the namespace you used for MATLAB Online Server:

```
kubectl describe pod pod_name –n your-namespace
```

In the information returned, look at the Events section. The following table lists the most likely reasons for the ImagePullBackOff error.

Message Type	Sample Error Message	Reason
Invalid container image or no pull access for a private image	`Error response from daemon: repository foobartest4 not found: does not exist or no pull access`	This means either that the specified registry name is nonexistent, or you do not have access to the container. Make sure you have the correct name. If the name is correct, then check that the container registry for this image does not require authentication or, if it requires authentication, make sure that you have the secret properly configured. As a test, use the following command to try to pull the same image from your local machine: `docker pull <docker-image-name:docker-image-tag>`

Invalid container image tag	`Warning  Failed 10m (x4 over 12m) kubelet, mos-gar-3-pool-1-9781becc-bdb3  Failed to pull image "redis:foobar": rpc error: code = Unknown desc = Error response from daemon: manifest for redis:foobar not found`	This means that a specified image cannot be found. To test if the image has an appropriate tag, try to pull the image locally: `docker pull <docker-image-name:docker-image-tag>`

8.5.2.2 CrashLoopBackOff

This pod status means that your pod is starting, crashing, starting again, and then crashing again.

To get more information about the problem, run the following command, replacing *pod_name* with the name of the pod that is stuck and *your-namespace* with the namespace you used for MATLAB Online Server:

kubectl describe pod *pod_name* –n *your-namespace*

In the information returned, look at the Events section, which is the main part of the log that describes what is happening with the pod. Examples of output for the Events section can be found in Check Pod Status.

Next, look at the logs by running the following command, replacing *pod_name* with the name of the pod that is stuck:

```
kubectl logs pod_name
```

For a multi-container pod (for example, the MATLAB pod), run the command with the container name:

```
kubectl logs pod_name -c container_name
```

Logs can help you find a reason why the container is crashing. Look at the logs for Java exception or error messages. See Get Help for assistance.

8.5.2.3 Pending

To get more information about why the pod is in a pending state, run the following command, replacing *pod_name* with the name of the pod that is stuck and *your-namespace* with the namespace you used for MATLAB Online Server:

```
kubectl describe pod pod_name –n your-namespace
```

The pod might be in a pending state because there is not enough CPU, not enough memory, or not enough CPU *and* memory.

If there is not enough CPU, for example, you might find the following entry in the log:

```
...

...

Events:

  Type     Reason            Age                From             Message

  ----     ------            ----               ----               ----

  Warning  FailedScheduling  2s (x6 over 11s) default-scheduler  0/4
nodes are available: 4 Insufficient cpu
```

Possible conditions:

- You have requested more CPU than *any* of the nodes has. For example, if each node in the two-node cluster has 2 CPU cores and you request 4 CPU cores, even though, in total, the cluster capacity is 4 CPU cores, a single node does not meet the request of 4 CPU cores.

  Every pod needs to be scheduled on a single node with 4 CPU cores and because all nodes have only 2 CPU cores, the request cannot be met. Even if you turn on more nodes in your cluster, Kubernetes still is not able to schedule your request.

- There is no more capacity in the cluster for the CPU cores you have requested. For example, if each node in the two-node cluster has 1 CPU core and you request 1 CPU core, then two pods can be scheduled on each node because the requested CPU matches the node's capacity.

  However, if a third pod requests 1 CPU core, there is not enough CPU in the cluster, and you get the FailedScheduling state. If this happens, you can turn on more nodes (with 1 CPU core) in the cluster, and your pod can be scheduled.

8.5.2.4 ContainerCreating

When a pod is just starting, ContainerCreating is a normal state. However, it is not normal for a pod to be stuck in the ContainerCreating state for longer than a few seconds (approximately 10 to 20 seconds).

To get more information about the problem, run the following command, replacing *pod_name* with the name of the pod that is stuck and *your-namespace* with the namespace you used for MATLAB Online Server:

```
kubectl describe pod pod_name –n your-namespace
```

In the information returned, look at the Events section to find a possible reason for the pod to continue to be in the ContainerCreating state.

8.5.2.5 Get Help

If none of the above issues described in this section help you to resolve the issue, try the following:

- See MATLAB Pod Issues.
- Contact MathWorks Support.
  1. Run the following kubectl commands, replacing *license_pod_name* with the name of the pod that is stuck and *your-namespace* with the namespace you used for MATLAB Online Server:

     ```
     kubectl describe pod license-pod-name --namespace your-namespace
     ```

     ```
     kubectl logs license-pod-name --namespace your-namespace
     ```

  2. Keep the output nearby to discuss with a support technician.

8.5.3 Evicted or Terminated Pods

8.5.3.1 How Pods Become Evicted

When available compute resources are low, some pods can get into an Evicted state. When resources become available on the node, these pods are automatically rescheduled.

To see pod status, run the following command, replacing *your-namespace* with the namespace you used for MATLAB Online Server:

```
kubectl get pods -n your-namespace
```

8.5.3.2 How to Clean Up Evicted Pods

- Run a Cron job to delete Evicted pods at regular intervals.
- Manually delete evicted pods using the following command, replacing *your-namespace* with the namespace you used for MATLAB Online Server:

  ```
  kubectl get pods -n <namespace> | grep Evicted | \

    awk '{print $1}' | xargs kubectl \
    delete pod -n your-namespace
  ```

8.5.3.3 How Pods Become Terminated

A Terminated state indicates that the container completed its execution and has stopped running. A pod enters the Terminated state when it has successfully completed execution or when it has failed.

Kubernetes cleans up Terminated pods when the number of pods exceeds the configured threshold, as determined by terminated-pod-gc-threshold in the kube-controller-manager. For a single-node installation, terminated-pod-gc-threshold is set to 10.

To see the status of pods, run the following command, replacing *your-namespace* with the namespace you used for MATLAB Online Server:

```
kubectl get pods -n your-namespace
```

8.5.3.4 How to Clean Up Terminated Pods

You can clean up pods in a Terminated state using one of these methods:

- Run a Cron job to delete Terminated pods at regular intervals.
- Manually delete terminated pods using the following command, replacing *your-namespace* with the namespace you used for MATLAB Online Server:

```
kubectl get pods -n <namespace> | grep Terminated | \
awk '{print $1}' | xargs kubectl \
delete pod -n your-namespace
```

Occasionally, manual deletion does not actually delete the terminated pod. In this case, you might have to forcibly delete the pod. Run the following command, replacing *your-namespace* with the namespace you used for MATLAB Online Server:

```
kubectl get pods -n <namespace> | grep Terminated | \
   awk '{print $1}' | xargs kubectl \
   delete pod -n your-namespace --force --grace-period=0
```

For more about the kubectl command, see the kubectl Cheat Sheet on kubernetes.io.

8.5.4 Out-of-Resource Conditions

Depending on how the cluster resources are configured, under certain situations, the cluster might be out of physical resources, which can affect pod status.

To see pod status, run the following command, replacing *your-namespace* with the namespace you used for MATLAB Online Server:

```
kubectl get pods -n your-namespace
```

8.5.4.1 How Out-of-Resource Condition Can Occur

Some pod statuses that can indicate that the cluster is out of resources include Pending, Evicted (see Evicted or Terminated Pods), and ContainerCreating.

Kubernetes can proactively monitor for and prevent total starvation of a compute resource. If this situation occurs during pod creation, Kubernetes can reclaim the starved resource by proactively failing one or more pods. The pod status then goes from Running to Evicted, and the new pods get stuck in the ContainerCreating state.

8.5.4.2 Possible Solution 1. Clean Up Unused Images

An out-of-resource condition can occur if you update the MATLAB image too many times without cleaning up unused images. For example, if the ephemeral storage of the EC2 instance (node) in AWS is approximately 100 GB, and you spin up a single-node MATLAB Online Server cluster successfully with all pods in the Running state, most capacity on the node is occupied by the MATLAB image (approximately 27 GB) for a full MATLAB install.

If you happen to update the image in a few weeks and perform a MATLAB Pool update, the node downloads yet another approximately 27 GB image (assuming it is an image with a different name). If you do this a few more times without cleaning up unused images, you start seeing that some pods are getting into an Evicted state and the pod you are trying to start (MATLAB pod in this case) is stuck in the ContainerCreating state because it was able to pull the image but did not have enough resources (memory) to start the container.

Make sure unused images from the node are periodically cleaned up. Run the following command to see the images on the node:

```
docker images
```

Then, use either of the following commands to remove an unused image:

```
docker rmi <image-id>
```

Or

```
docker rmi <image:tag>
```

8.5.4.3 Possible Solution 2. Provision More Nodes or Set Explicit Resource Limits

If resource limits for CPU and memory are not set, users can inadvertently exploit all available resources.

For example, assume that the MATLAB Pool pod is configured with no resource limits (default) as shown:

```
matlabResources:
  requests:
    cpu: 200m
    memory: 2Gi
  limits:
    #cpu: <cpu-limit-here>
    #memory: <memory-limit-here>
```

Under these conditions, when a MATLAB Online user performs an operation that maxes out the resources on the node, other end users cannot sign in to MATLAB Online, as the pod is stuck in the ContainerCreating state. Because there is a lack of resources, Kubernetes starts evicting some pods, putting them in the Evicted state.

8.5.4.3.1 Possible Action 1: Provision More Nodes

Provision more nodes on the cluster to meet the resource needs of the cluster.

8.5.4.3.2 Possible Action 2: Set Explicit Resource Limits

Set resource limits appropriately for the MATLAB Pool pod so that users cannot exploit all the resources.

1. Update the matlab-pool override file at <overrides/matlab-online-server/mathworks/matlab-pool.yaml>. Uncomment the limits section and update with appropriate limit values.
2. Run the following mosadm command to update the node:

   ```
   mosadm upgrade matlab-pool
   ```

8.5.5 **License Service Issues**

MATLAB Online Server requires you to have valid licenses for both MATLAB Online Server and MATLAB before you install either. After you have completed installation with valid licenses, your users can sign in successfully to MATLAB Online.

8.5.5.1 License Service Pod State is Running but Users Cannot Connect to MATLAB Online

When the services are running successfully and they can connect with the FLEXlm license server, then the status of pods is Running.

However, if the services are having issues connecting with FLEXlm, the status of the pods is still shown as Running. This is expected, and it means that the MATLAB Online Server is running successfully but is not authorized because it is unable to connect to the network license manager.

```
ubuntu@ubuntu:~/matlab_online_server$ kubectl get pods --namespace
your-namespace

NAME                                                          READY
STATUS    RESTARTS   AGE

your-namespace-core-ui-core-ui-5bd9b7c88b-fqbl4                    1/1
Running   0     111s

your-namespace-authnz-authnz-7c7999589c-x74d9                      1/1
Running   0     111s

your-namespace-gateway-669dd7865f-t6l74                            1/1
Running   0     110s

your-namespace-license-548f7b6655-8skdt                            1/1
Running   0     108s

your-namespace-matlab-pool-8c679f76-c4h4n                          4/4
Running   0     107s

your-namespace-matlab-pool-8c679f76-cckkp                          4/4
Running   0     107s

your-namespace-matlab-pool-ui-579fc8ff49-2bz25                     1/1
Running   0     107s

your-namespace-nginx-ingress-controller-kcwwd                      1/1
Running   0     104s

your-namespace-nginx-ingress-default-backend-f78c5f7b8-dvxm8   1/1
Running   0         104s

your-namespace -resource-resource-894447df7-r9ngb                   1/1
Running  0     103s
```

If the license service is not operational, other services also do not work as expected. This is because the license service checks out the MATLAB Online Server license, which in turn allows all other services to run successfully.

8.5.5.1.1 Get More Information

Run the following command to get more information, replacing *license-pod-name* with the name of your license pod and *your-namespace* with the namespace you used for MATLAB Online Server:

```
kubectl logs license-pod-name --namespace your-namespace
```

If you get output similar to the following display, then it is likely that the license service is unable to check out the MATLAB Online Server license.

```
00000006 2020-Jan-28 21:10:55.104617 0x00007fa8d8fd5700 mos::license I
Checking out license

00000007 2020-Jan-28 21:10:55.104626 0x00007fa8d8fd5700 mos::license I
Server: 27000@flexlm

00000008 2020-Jan-28 21:11:15.237586 0x00007fa8d8fd5700
startup::license_manager I enter checkout for feature: MATLAB_Online_Server

The server is running.

Error while trying to check out the license: , will retry in 30 seconds.

00000009 2020-Jan-28 21:11:28.314607 0x00007fa8d8fd5700 mos::license I
Registered service

00000010 2020-Jan-28 21:11:56.040342 0x00007fa8d203c700 mos::license I Check
license request mathworks-gateway-669dd7865f-t6l74

00000011 2020-Jan-28 21:11:58.430979 0x00007fa8c71fa700 mos::license I
Checking out license

00000012 2020-Jan-28 21:11:58.431050 0x00007fa8c71fa700 mos::license I
Server: 27000@flexlm

00000013 2020-Jan-28 21:12:03.866191 0x00007fa8d303e700 mos::license I Check
license request from mathworks-authnz-authnz-7c7999589c-x74d9

00000014 2020-Jan-28 21:12:04.565262 0x00007fa8c71fa700
startup::license_manager I enter checkout for feature: MATLAB_Online_Server

00000015 2020-Jan-28 21:12:04.869012 0x00007fa8d283d700 mos::license I Check
license request from mathworks-resource-resource-894447df7-r9ngb

Error while trying to check out the license: ^^, will retry in 30 seconds.
```

8.5.5.1.2 Possible Issues

- FLEXlm server listening on ports other than 27000/27001

  Network policy is enabled by default for the license service, which allows communicating with FLEXlm only on port 27000 and with the MLM daemon on 27001. The default policy assumes that the license file has the following SERVER lines:

  ```
  SERVER <HOSTID> ID=0 27000
  DAEMON MLM "<FLEXLM_INSTALL_LOCATION>/etc/MLM" port=27001
  ```

If your FLEXlm server is listening on ports other than 27000 and the MLM daemon on other than 27001, update the license service override file and restart the pod:

```
networkPolicy:
  enabled: true
  egress:
      ports:
      - port: <flexlm_port_here>
        protocol: TCP
```

- FLEXlm server is using an incorrect or an expired license file or FLEXlm server is misconfigured

  Check the FLEXlm server logs to see details about the reason for failure. Try to connect to the FLEXlm server from a local machine using the following commands:

  ```
  nc -vz <FelxLM_DNS_NAME> 27000
  nc -vz <FelxLM_DNS_NAME> 27001
  ```

  In cases where the FLEXlm server is using an expired license file:

- Update the license file. See Update Network License on the MathWorks website.
- Set up both a listener on the FLEXlm log file and an alert, so that you are notified when the license is about to expire.
- If the license service pod is not in the Running state, see License Service Issues.
- See MATLAB Pod Issues.

8.5.5.2 Get Help

If you are unable to resolve the issue with these resources, contact MathWorks Support and have ready the output from the following kubectl commands:

```
kubectl describe pod <license-pod-name> --namespace <namespace>

kubectl logs <license-pod-name> --namespace <namespace>
```

8.6 Fix Security Exception Error When Accessing Help

8.6.1 Problem

Your end user runs a doc command at the MATLAB command prompt, for example:

```
>> doc plot
```

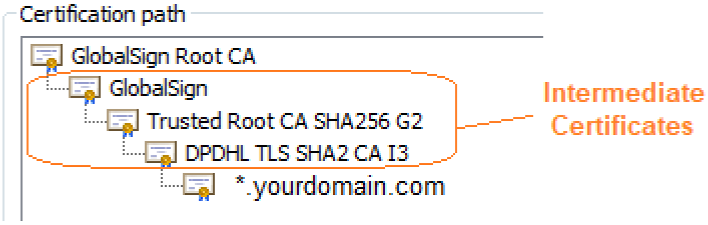
They see an error similar to the following:

```
Error using doc>displayDocPage (line 171)
Java exception occurred:
com.mathworks.webservices.client.core.MathWorksClientException: unable
to connect to host sun.security.validator.ValidatorException: PKIX
path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to
find valid certification path to requested target
. . .
```

8.6.2 Solution

This exception occurs when the Local CA certificate store is missing one or more intermediate CA certificates. For the client to validate the server, Root CA certificates must be available in the local certificate store. Without these intermediate Certificates, the client considers the secure connection untrusted and, in most cases, rejects it.

The figure below shows an example of a certificate path that can be viewed from the Chrome browser. In this example, the Certificate of *.yourdomain.com has "GlobalSign Root CA" as root and three intermediate certificates: GlobalSign, Trusted Root CA SH256 G2, and DPDHL TLS SHA2 CA I3.



To resolve missing certificates:

1. Append the missing intermediate CA certificates to your public key file (for example, publicKey.pem).

2. Open your publicKey.pem file, and then append only the missing intermediate certificates to it in the reverse order. Do not append the root certificate to this file.

For this example, the file publicKey.pem must contains the intermediate certificates in this order:

-----BEGIN CERTIFICATE-----

PDHL TLS SHA2 CA I3

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

```
        Trusted Root CA SH256 G2
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
        GlobalSign
-----END CERTIFICATE-----
```

9 Server Maintenance

9.1 Smoke Testing MATLAB Online Server

Smoke tests reveal simple failures severe enough to impact operations. The smoke test tool that comes with MATLAB Online Server runs a set of tests against a MATLAB Online Server implementation. The tool returns information on installation setup, Kubernetes elements, and the MATLAB Online Server microservices. The information provided by these smoke tests can help focus troubleshooting strategies for MATLAB Online Server.

You can execute smoke tests as the option `run-smoke-tests` to the mosadm command. The following example shows the default execution of smoke tests without any additional arguments:

./mosadm run-smoke-tests

9.1.1 Prerequisites

- Single Node
  - Set and export your test username and password environment variables MOS_TEST_USERNAME and MOS_TEST_PASSWORD. Their default values are *admin* and *password*, respectively. The user name and password should be valid credentials for signing in to your MATLAB Online Server session via the browser.
  - If you have installed MATLAB Online Server using mosadm, mosadm applies the correct Kube-Config and properly sets the environment variable KUBECONFIG.
- Managed Kubernetes (EKS, GKE, AKS, …)
  - Set and export your test user name and password environment variables MOS_TEST_USERNAME and MOS_TEST_PASSWORD. Their default values are *admin* and *password*, respectively.
  - Make sure that you can access your Kubernetes cluster via kubectl.
  - Your ~/.kube/config should point to your cluster and namespace Kube-Config file, or your KUBECONFIG environment variable should be set to the right path of your Kube-Config file. Note that KUBECONFIG has priority over ~/.kube/config.

9.1.2 Arguments for Smoke Test Option

You can pass any of the arguments described in this topic to the run-smoke-tests option, alone or in conjunction with another argument.

-verbose

To get more details on the individual smoke tests, specify this argument. This argument is useful when some tests are failing, so that you can get more information about the failure.

-no-certificate-check

If the certificate deployed at MATLAB Online Server does not match its domain, many tests will fail. You can specify this argument to ignore this certificate mismatching.

-no-secure-http

If no certificate is deployed at MATLAB Online Server, you can specify this argument to run tests that require HTTP calls using only non-secure http. This argument can be helpful in checking quickly if MATLAB Online Server is deployed properly before deploying the appropriate certificate in place. If this argument is specified, then the argument no-certificate-check is ignored.

9.1.2.1 Examples

9.1.2.1.1 No arguments

```
./mosadm run-smoke-tests
```

```
2020/05/28 18:18:21 Smoke-Tests
2020/05/28 18:18:21   Commands
2020/05/28 18:18:22     docker : PASS
2020/05/28 18:18:22     kubectl : PASS
2020/05/28 18:18:22     helm : PASS
2020/05/28 18:18:22   Commands : PASS
2020/05/28 18:18:22
2020/05/28 18:18:22   Properties
2020/05/28 18:18:22     DomainBaseLookup : PASS
2020/05/28 18:18:22     DomainBaseSecurelyReachable : PASS
2020/05/28 18:18:22     DomainBaseValidateCertificate : PASS
2020/05/28 18:18:22   Properties : PASS
2020/05/28 18:18:22
2020/05/28 18:18:22   Docker
2020/05/28 18:18:22     DockerRootDirSpace : PASS
2020/05/28 18:18:22   Docker : PASS
2020/05/28 18:18:22
2020/05/28 18:18:22   Kubernetes
2020/05/28 18:18:22     GetKubernetesClient : PASS
2020/05/28 18:18:22     CheckKubernetesNamespace : PASS
2020/05/28 18:18:22     GetPodsList : PASS
2020/05/28 18:18:23     MOS-Core-ServicesPodsAreReady : PASS
2020/05/28 18:18:23   Kubernetes : PASS
2020/05/28 18:18:23
2020/05/28 18:18:23   MOS
2020/05/28 18:18:23     AuthNZ
2020/05/28 18:18:23       UserNamePasswordAuthentication : PASS
2020/05/28 18:18:23     AuthNZ : PASS
2020/05/28 18:18:23
2020/05/28 18:18:23     CoreUI
2020/05/28 18:18:23       CheckFavoriteIcon : PASS
2020/05/28 18:18:23     CoreUI : PASS
2020/05/28 18:18:23
2020/05/28 18:18:23     Gateway
2020/05/28 18:18:23       CreateMapping : PASS
2020/05/28 18:18:23     Gateway : PASS
2020/05/28 18:18:23
2020/05/28 18:18:23     License
```

```
2020/05/28 18:18:23       BuildPortForwardConfig : PASS
2020/05/28 18:18:23       CheckMosLicense : PASS
2020/05/28 18:18:23     License : PASS
2020/05/28 18:18:23
2020/05/28 18:18:23     MATLABPoolUI
2020/05/28 18:18:23       CheckRemoteProxyScript : PASS
2020/05/28 18:18:23     MATLABPoolUI : PASS
2020/05/28 18:18:23
2020/05/28 18:18:23     Resource
2020/05/28 18:18:23       AcquireResource : PASS
2020/05/28 18:18:23     Resource : PASS
2020/05/28 18:18:23
2020/05/28 18:18:23     Help
2020/05/28 18:18:23       StaticReleaseNote : PASS
2020/05/28 18:18:23       StaticReleaseNoteVersioned : PASS
2020/05/28 18:18:23       DynamicSearchSuggest : PASS
2020/05/28 18:18:23       DynamicSearchJson : PASS
2020/05/28 18:18:23     Help : PASS
2020/05/28 18:18:23
2020/05/28 18:18:23   MOS : PASS
2020/05/28 18:18:23
2020/05/28 18:18:23 Smoke-Tests : PASS
2020/05/28 18:18:23
```

9.1.2.1.2 With Verbose Argument

```
./mosadm run-smoke-tests  --verbose
```

```
2020/05/28 18:20:15 Smoke-Tests
2020/05/28 18:20:15   Commands
2020/05/28 18:20:15     docker
2020/05/28 18:20:15       docker version : Client:
                     Version:          18.09.2
                     API version:      1.39
                     Go version:       go1.10.6
                     Git commit:       6247962
                     Built:            Sun Feb 10 04:13:52 2019
                     OS/Arch:          linux/amd64
                     Experimental:     false

                    Server: Docker Engine - Community
                     Engine:
                      Version:          18.09.2
                      API version:      1.39 (minimum version 1.12)
                      Go version:       go1.10.6
                      Git commit:       6247962
                      Built:            Sun Feb 10 03:42:13 2019
                      OS/Arch:          linux/amd64
                      Experimental:     false
2020/05/28 18:20:15     docker : PASS
2020/05/28 18:20:15
2020/05/28 18:20:15     kubectl
2020/05/28 18:20:15       kubectl version : Client Version: version.Info{Major:"1", Minor:"16",
GitVersion:"v1.16.2", GitCommit:"c97fe5036ef3df2967d086711e6c0c405941e14b", GitTreeState:"clean",
BuildDate:"2019-10-15T19:18:23Z", GoVersion:"go1.12.10", Compiler:"gc", Platform:"linux/amd64"}
                    Server Version: version.Info{Major:"1", Minor:"14+", GitVersion:"v1.14.9-eks-502bfb",
GitCommit:"502bfb383169b124d87848f89e17a04b9fc1f6f0", GitTreeState:"clean", BuildDate:"2020-02-
07T01:31:02Z", GoVersion:"go1.12.12", Compiler:"gc", Platform:"linux/amd64"}
2020/05/28 18:20:15     kubectl : PASS
```

```
2020/05/28 18:20:15
2020/05/28 18:20:15      helm
2020/05/28 18:20:15        helm version : Client: &version.Version{SemVer:"v2.15.1",
GitCommit:"cf1de4f8ba70eded310918a8af3a96bfe8e7683b", GitTreeState:"clean"}
                    Server: &version.Version{SemVer:"v2.15.0",
GitCommit:"c2440264ca6c078a06e088a838b0476d2fc14750", GitTreeState:"clean"}
2020/05/28 18:20:15      helm : PASS
2020/05/28 18:20:15
2020/05/28 18:20:15    Commands : PASS
2020/05/28 18:20:15
2020/05/28 18:20:15    Properties
2020/05/28 18:20:15      DomainBaseLookup
2020/05/28 18:20:15          DomainBase : mos-test.mathworks.com
2020/05/28 18:20:15      DomainBaseLookup : PASS
2020/05/28 18:20:15      DomainBaseSecurelyReachable : PASS
2020/05/28 18:20:16      DomainBaseValidateCertificate : PASS
2020/05/28 18:20:16    Properties : PASS
2020/05/28 18:20:16
2020/05/28 18:20:16    Docker
2020/05/28 18:20:16      DockerRootDirSpace
2020/05/28 18:20:16        Docker Root Dir Disk Space: 15312 MB
2020/05/28 18:20:16      DockerRootDirSpace : PASS
2020/05/28 18:20:16    Docker : PASS
2020/05/28 18:20:16
2020/05/28 18:20:16    Kubernetes
2020/05/28 18:20:16      GetKubernetesClient : PASS
2020/05/28 18:20:16
2020/05/28 18:20:16      CheckKubernetesNamespace
2020/05/28 18:20:16        namespace : mathworks
2020/05/28 18:20:16      CheckKubernetesNamespace : PASS
2020/05/28 18:20:16
2020/05/28 18:20:16      GetPodsList
2020/05/28 18:20:16        Pods List
2020/05/28 18:20:16          mathworks-matlab-pool-helpsearch-9dd565f48-crrnk : READY
2020/05/28 18:20:16          mathworks-nginx-ingress-controller-77c7d8b786-45hcv : READY
2020/05/28 18:20:16          mathworks-resource-resource-86556c6697-lhks6 : READY
2020/05/28 18:20:16          mathworks-test-environment-ldap-7f7cb88649-d72w8 : READY
2020/05/28 18:20:16          mathworks-authnz-authnz-68df77576b-8m52c : READY
2020/05/28 18:20:16          mathworks-license-5477b95664-ffzj5 : READY
2020/05/28 18:20:16          mathworks-matlab-pool-ui-57748ddd7d-jfghl : READY
2020/05/28 18:20:16          mathworks-test-environment-flexlm-7c75f595-db7c4 : READY
2020/05/28 18:20:16          mathworks-test-environment-nfs-5847fb79dd-44vzg : READY
2020/05/28 18:20:16          mathworks-core-ui-core-ui-86747b4d78-n86nc : READY
2020/05/28 18:20:16          mathworks-matlab-pool-6b6b6d88b-hr4x5 : READY
2020/05/28 18:20:16          mathworks-matlab-pool-apparmor-loader-2p5zm : READY
2020/05/28 18:20:16          mathworks-nginx-ingress-default-backend-686d46f787-8hwjz : READY
2020/05/28 18:20:16          mathworks-gateway-5b7cb59bbd-2zjzd : READY
2020/05/28 18:20:16          mathworks-matlab-pool-6b6b6d88b-wjssb : READY
2020/05/28 18:20:16      GetPodsList : PASS
2020/05/28 18:20:16
2020/05/28 18:20:16      MOS-Core-ServicesPodsAreReady
2020/05/28 18:20:16        MOS Core Services
2020/05/28 18:20:16          authnz : READY 1/1
2020/05/28 18:20:16          core-ui : READY 1/1
2020/05/28 18:20:16          gateway : READY 1/1
2020/05/28 18:20:16          license : READY 1/1
2020/05/28 18:20:16          matlab-pool : READY 2/2
2020/05/28 18:20:16          matlab-pool-helpsearch : READY 1/1
2020/05/28 18:20:16          matlab-pool-ui : READY 1/1
2020/05/28 18:20:16          resource : READY 1/1
```

```
2020/05/28 18:20:16     MOS-Core-ServicesPodsAreReady : PASS
2020/05/28 18:20:16   Kubernetes : PASS
2020/05/28 18:20:16
2020/05/28 18:20:16   MOS
2020/05/28 18:20:16     AuthNZ
2020/05/28 18:20:16       UserNamePasswordAuthentication : PASS
2020/05/28 18:20:16     AuthNZ : PASS
2020/05/28 18:20:16
2020/05/28 18:20:16     CoreUI
2020/05/28 18:20:16       CheckFavoriteIcon : PASS
2020/05/28 18:20:16     CoreUI : PASS
2020/05/28 18:20:16
2020/05/28 18:20:16     Gateway
2020/05/28 18:20:16       CreateMapping : PASS
2020/05/28 18:20:16     Gateway : PASS
2020/05/28 18:20:16
2020/05/28 18:20:16     License
2020/05/28 18:20:16       BuildPortForwardConfig : PASS
2020/05/28 18:20:16       CheckMosLicense : PASS
2020/05/28 18:20:16     License : PASS
2020/05/28 18:20:16
2020/05/28 18:20:16     MATLABPoolUI
2020/05/28 18:20:16       CheckRemoteProxyScript : PASS
2020/05/28 18:20:16     MATLABPoolUI : PASS
2020/05/28 18:20:16
2020/05/28 18:20:16     Resource
2020/05/28 18:20:16       AcquireResource : PASS
2020/05/28 18:20:16     Resource : PASS
2020/05/28 18:20:16
2020/05/28 18:20:16     Help
2020/05/28 18:20:16       StaticReleaseNote : PASS
2020/05/28 18:20:16       StaticReleaseNoteVersioned : PASS
2020/05/28 18:20:16       DynamicSearchSuggest : PASS
2020/05/28 18:20:16       DynamicSearchJson : PASS
2020/05/28 18:20:16     Help : PASS
2020/05/28 18:20:16
2020/05/28 18:20:16   MOS : PASS
2020/05/28 18:20:16
2020/05/28 18:20:16 Smoke-Tests : PASS
```

9.1.3 Addressing Failures

Before proceeding with troubleshooting failures shown by the smoke tests, make sure that the prerequisites stated in the MATLAB Online Server Administrator Guide are satisfied.

Examine the results of the smoke tests and determine if all tests passed or if there are failures that you must investigate.

9.1.3.1 Pass

If you get a **PASS** result on all smoke tests, then you have properly deployed MATLAB Online Server, and you are ready to access it from your browser.

9.1.3.2 Fail

If you get a **FAIL** for one ore more sub smoke tests, then you must find the root cause of their failures before proceeding further.

The smoke tests are divided into several parts, where each part has a subset of tests that target a specific part of configuration or behavior of your local setup, Kubernetes configuration, or MATLAB Online Server setup.

If you get a FAIL on any sub smoke test, re-run the smoke tests with the argument *–verbose*. This argument gives you more details on the failures.

9.1.3.3 Commands Tests

The commands tests check if the commands *docker*, *kubectl,* and *helm* run properly from the machine on which you are running the *mosadm* command. If the –verbose option is enabled, you get more details on their versions.

If any of the subtests fails, you can manually check if the related command (*docker*, *kubect,l* or *helm*) is installed on your machine and properly working.

9.1.3.4 Properties Tests

The properties checks focus on tests related to your MATLAB Online Server *domain*.

- *DomainBaseLookup* checks if your domain is resolvable to an IP address.
- *DomainBaseSecurelyReachable* checks if your domain is reachable securely via https.
- *DomainBaseValidateCertificate* checks if your domain SSL certificate matches your domain name.

While the MATLAB Online Server installation instructions specify that you have a valid SSL certificate that matches your MATLAB Online Server domain name, you may not have it initially ready. However, you can still run the smoke tests.

If you don’t have a valid SSL certificate or you have one but it does not match your domain (for example, you may have a self-signed certificate), you can have the smoke test bypass the certificate requirement by passing the no-secure-http and no-certificate arguments to the smoke tests option:

```
mosadm run-smoke-tests -no-secure-http -no-certificate-check
```

9.1.3.5 Docker Tests

The Docker tests check properties related your local docker environment. The test checks for the Docker Root Directory disk space (refer to System Requirements for details on disk space requirements).

If the available disk space under this directory disk partition is low, then that might be a root cause for some deployment failures. Consider increasing this disk space, or re-install or re-configure Docker to use a Root Directory under a partition with more disk space.

9.1.3.6 Kubernetes Tests

The Kubernetes tests run a set of tests related to your Kubernetes cluster.

- *GetKubernetesClient* checks if the test can communicate with your Kubernetes cluster. If this test fails, then make sure that your KUBECONFIG (or ~/.kube/config) is set properly and you can access it successfully from your installation machine using the kubectl command.
- *CheckKubernetesNamespace* verifies that your configured namespace, which defaults to mathworks, exists under your Kubernetes cluster. If the configured namespace is not present

under your Kuberenetes cluster, then your MATLAB Online Server won't be deployed properly, or will be deployed but under the wrong namespace.

- *GetPodsList* displays all pods under your configured namespace. This is mainly an informative test that displays all existing pods under your Kubernetes namespace.
- *MOS-Core-ServicesPodsAreReady* lists only the MATLAB Online Server core service pods. If a pod is running properly, the test shows the status READY next to the service. Otherwise, the test shows which underlying container is failing, along with any error messages that apply.

9.1.3.7 MATLAB Online Server

MATLAB Online Server runs targeted tests against each MATLAB Online core service.

- *AuthNZ* checks if the AuthNZ service can authenticate users via their usernames and passwords. Under Prerequisites, you set two environment variables: MOS_TEST_USERNAME and MOS_TEST_PASSWORD. These variables store the valid username and password, respectively, for accessing your MATLAB Online Server from the browser. These variables default to *admin* and *password*.

  If this AuthNZ test fails, even if those environment variables are set properly, then check if the License test is failing. If this is the case, first fix the License service and then re-run the smoke tests. If the License test passes and the AuthNZ test is still failing, then check the AuthNZ service logs for further troubleshooting.
- *CoreUI*, *Gateway*, *MATLABPoolUI* and *Resource* run a set of internal MATLAB Online Server tests. Like *AuthNZ*, if any of these services tests fails, first check if the License service is passing its test before troubleshooting any of them.
- *License* checks if the MATLAB Online Server License service is responding to requests. If this test shows the status PASS but other MATLAB Online Server services are still failing their tests, check if your FlexLM server is accessible, via its ports, from your Kubernetes pods. The *netcat* (*nc*) can be helpful in checking if a *server* is accessible via specific *port*.

Also check if your license file (license.dat) contains your MATLAB licenses as well as your MATLAB Online Server license.

- *Help* tests if the Help content is accessible by MATLAB Online Server users. Even if these tests are failing, it does not have any impact on the remaining MATLAB Online Server services.

If you get many MATLAB Online Server services smoke tests failing, start by troubleshooting the License service and then proceed further with the remaining services.

As a general best practice, after fixing each single failed sub smoke test, re-run all smoke tests. Some services depend on others, and fixing one of them, like the License, can in many cases fix other failures.

If you continue to see errors in some sub smoke tests of some MATLAB Online Server services, check their logs (see Log Shipping), as these files often provide useful information on the root causes of such failures.

If you are still experiencing failures after applying the troubleshooting suggestions, prepare the following items, and then contact MathWorks Support:

- Save the *logs* folder, generated by the smoke tests, as a zip file.

- Run the command “sudo ./mosadm gather-system-info” which generates the file install.log.

9.2 Parallel Computing Toolbox and MATLAB Distributed Computing Server—Administrators

9.2.1 Overview

This topic covers the tasks you, as the MATLAB Online Server administrator, need to perform so that your end users can use Parallel Computing Toolbox™ and MATLAB® Distributed Computing Server™.

By default, the Helm charts that define the MATLAB Online Server cluster enable support for Parallel Computing Toolbox and the automatic creation of local parpool clusters. This behavior can be changed by editing the matlab-pool.yaml override file.

To enable access to MATLAB Distributed Computing Server Clusters for MATLAB Online Server users, you must add a NetworkPolicy egress rule to the matlab-pool.yaml file in the override’s folder.

9.2.2 Configure Support in MATLAB Pool

There are five configuration variables that can be set in the matlab-pool.yaml file of your MATLAB Online Server install overrides directory.

By default, the values are:

```
pct:
  enabled: true
  defaultPoolSize: 2
  defaultPoolAutoCreate: true

matlabProcessConfiguration:
  enableLocalParCluster: true
```

The above set of values configure the installation so that:

- Parallel Computing Toolbox is available for use by users of MATLAB Online Server.
- The default pool size of the Parallel Computing Toolbox pools is 2.
- The default parcluster (via parpool) is 'local'.
- A parpool with the default will automatically be created if the user runs MATLAB code containing 'parfor' or 'spmd'.

Additionally, the MATLAB Process is configured to enable the use of local MATLAB workers on the same Kubernetes node as the MATLAB Docker container is running.

Given these values, it is possible for you to configure the MATLAB instances to use an on-premise cluster but not enable the use of local MATLAB workers on the Kubernetes nodes running MATLAB by adding these settings to the matlab-pool.yaml override file.

```
pct:
  enabled: true

matlabProcessConfiguration:
  enableLocalParCluster: false
```

9.2.3 Enable Use of On-Premise MJS or HPC Clusters

MATLAB Online Server is installed with a set of network policies that restrict the egress of network connections from nodes that the MATLAB Docker containers are running on. Consequently, to enable the use of MJS or access to other HPC job schedulers, you must update the matlab-pool.yaml overrides file. (For more information, see the overview documentation for How Parallel Computing Products Run a Job.)

The following YAML example demonstrates opening up egress for network connections to an on-premise host running the MathWorks MJS job scheduler with the IP address of 172.28.152.244. Other egress rules from the matlab-pool/values.yaml file have been copied in.

```
networkPolicy:
  enabled: true
  egress:
    # This first rule enables egress to all ports on the 172.28.152.244 host
    - to:
      - ipBlock:
      cidr: 172.28.152.244/32
    # This next rule is the rule for the other MOS internal nodes
    - to:
      - ipBlock:
      cidr: 0.0.0.0/0
      except:
        - 169.254.169.0/24
      ports:
        # allow dns access
        - port: 53
          protocol: TCP
        - port: 53
          protocol: UDP
        # allow standard http/https port access
        - port: 80
          protocol: TCP
        - port: 443
          protocol: TCP
        # allow access to the flexlm server port (ports controlled by license.dat)
        - port: 27000
          protocol: TCP
          - port: 27001
            protocol: TCP
        # allow access on a variety of nfs related ports (derived from portmap/lsof in nfs
server container)
        - port: 111
          protocol: UDP
        - port: 111
          protocol: TCP
        - port: 612
          protocol: UDP
        - port: 2049
          protocol: TCP
        - port: 2049
          protocol: UDP
        - port: 20048
          protocol: UDP
        - port: 20048
          protocol: TCP
        - port: 33668
          protocol: UDP
        - port: 33743
          protocol: TCP
```

```
- port: 38494
  protocol: UDP
- port: 36663
  protocol: TCP
- port: 38494
  protocol: UDP
- port: 40051
  protocol: TCP
- port: 52241
  protocol: UDP
```

9.2.4 End User Instructions

For the end user instructions, see Parallel Computing Toolbox and MATLAB Distributed Computing Server – End Users.

9.3 Database Setup with MATLAB Online Server

MATLAB Online end users can connect to a database to execute queries. MATLAB Online Server supports connections to JDBC-compliant databases; to learn more, see Getting Started with the Database Toolbox on the MathWorks website. As the administrator, update the configuration to allow proper communication to the database.

Before you begin, make sure you have set up MATLAB Database Toolbox as an add-on (see Install Add-Ons for MATLAB Online on the MathWorks website). In addition, you need the database host and port information so you can properly configure the back end.

9.3.1 Basic Configuration

MATLAB Online Server, by default, allows access to a minimum number of network ports to be used by the MATLAB process. To allow interactions with an external database, update the configuration YAML files to allow egress from the Kubernetes cluster to the desired resource.

1. Determine the required port for the database. Many database vendors have a default port, but it is possible this port is different in your environment. It is best to verify before beginning.

2. Update the overrides file matlab-pool.yaml file with the required port information for this database. Make your changes in the networkPolicy section of the yaml. See Configure Network Policies.

   An example matlab-pool.yaml configuration follows.

   ```
   networkPolicy:
     enabled: true
     egress:
       - ports:
         # allow MySQL access on standard port
         - port: 3306
           protocol: TCP
         - port: 3306
           protocol: UDP
         … <any additional port configurations>
   ```

3. Deploy the change to the cluster by issuing the following command, replacing *your-namespace* with the namespace used for MATLAB Online Server:

   ```
     mosadm upgrade matlab-pool -n your-namespace
   ```

4. Validate the change by viewing the Kubernetes network policy to look for your changes by issuing the following command, replacing *your-namespace* with the namespace used for MATLAB Online Server:

   ```
   kubectl describe networkPolicy -n your-namespace
   ```

9.3.2 Advanced Configuration

Provide a storage mount for all MATLAB Online users with read-only drivers. You can make version-specific drivers for your environment readily available to your end users. See NFS configuration to create a read-only mount that contains JDBC drivers for your MATLAB Online Server.

9.3.3 End User Instructions

See Database Access with MATLAB Online for instructions to give the end user for accessing their database.

10 Product Maintenance

10.1 Install Support Packages for MATLAB Online

A support package is a type of add-on that enables MATLAB users to use a MathWorks® product with specific third-party hardware and software. This topic describes how to install support packages in MATLAB Online so that they are accessible by all users.

10.1.1 Prerequisites

Before you begin, make sure you have done the following tasks:

- Install the Linux version of MATLAB from the Downloads page and launch MATLAB successfully.
- Learn what support packages are and how to manage them on the Linux version of desktop MATLAB. See Support Package Installation.
- Install and configure MATLAB Online Server.

10.1.2 Install Procedure

10.1.2.1 Step 1. Install MATLAB

Skip this step if you have already installed the Linux version of MATLAB as specified in Prerequisites.

1. Create an empty folder /MATLAB.
2. Install MATLAB R2020a in the folder /MATLAB with the toolboxes you want.

   Note that /MATLAB is `matlabroot`, not `/MATLAB/<releasename>`
3. Create a folder inside /MATLAB named SupportPackages:

   /MATLAB/SupportPackages

10.1.2.2 Step 2. Change Support Package Root

1. Launch MATLAB from /MATLAB/bin.
2. Change the support package root by issuing the following command:

   ```
   >>matlabshared.supportpkg.setSupportPackageRoot('/MATLAB/SupportPackages')
   ```
3. Restart MATLAB.

10.1.2.3 Step 3. Install Support Packages

1. Install any required support packages using the instructions in Get and Manage Add-Ons on the MathWorks website.

   Support packages that install third party applications on the machine are not supported.

2. To verify that the support packages are installed successfully, launch MATLAB and execute the following command:

   ```
   >> supportPackages = matlabshared.supportpkg.getInstalled
   ```

3. To verify that the support packages are available in MATLAB Online, launch MATLAB in the browser and execute the following command:

   ```
   >>addons = matlab.addons.installedAddons
   ```

4. Exit MATLAB.

5. Delete the licenses folder under /MATLAB.

10.1.2.4 Step 4. Build MATLAB Image

1. From the installation you just performed, build the MATLAB image using the following command:

   ```
   sudo ./mosadm build-matlab-image /MATLAB
   ```

   If you have a multi-node installation, you can retag the Docker images and push them to a remote Docker registry using the following command, replacing *your-registry* with the name of your Docker registry:

   ```
   sudo ./mosadm push-docker-images --docker-registry your-registry
   ```

2. Use mosadm to redeploy the MATLAB Pool pod using the following commands:

   ```
   mosadm undeploy matlab-pool
   ```

   ```
   mosadm deploy matlab-pool
   ```

10.2 Install Add-Ons for MATLAB Online

MATLAB add-ons encompass a wide variety of resources, including products, apps, support packages, and toolboxes. Add-ons extend the capabilities of MATLAB with functionality for additional tasks and applications. This topic describes how to install add-ons in MATLAB Online so that they are accessible by all users.

10.2.1 Prerequisites

Before you begin, make sure that you have done the following tasks.

- Learn what add-ons are and how to manage them on the Linux version of the MATLAB Desktop. For more information, see Get Add-Ons.
- Familiarize yourself with MATLAB Online Server installation and configuration. See Single-Machine Installation.

10.2.2 Install Procedure

10.2.2.1 Step 1. Create Docker Image with Add-Ons Installed

1. Go to Downloads on the MathWorks website. Install the Linux version of MATLAB R2020a in the folder /MATLAB.
2. Install the required add-ons using the Get and Manage Add-Ons instructions on the MathWorks Website.

   By default, MATLAB add-ons are installed in the following folder:

   ~/MATLAB Add-Ons
3. After you install the add-ons successfully, exit MATLAB.
4. Copy the folder where the add-ons were installed into the MATLAB installation directory /MATLAB.
5. Under /MATLAB, delete the licenses folder.
6. Using the following command, build the MATLAB image from the installation you just performed:

   ```
   sudo ./mosadm build-matlab-image /MATLAB
   ```

If you have a multi-node installation, you can retag the Docker images and push them to a remote Docker registry using the following command, replacing *your-registry* with the name of your Docker registry:

```
sudo ./mosadm push-docker-images --docker-registry your-registry
```

10.2.2.2 Step 2. Configure MATLAB Online Server to Use Installed Add-Ons

Each storage profile can optionally define an add-ons directory with the following YAML structure:

```
addons:
  directory: add-ons-directory
```

Ensure that *add-ons-directory* is available in the MATLAB container during runtime in one of the following manners:

- Inside the Docker container
- Through a special mount using the storage profile
- Chosen dynamically by the administrator

The benefits and drawbacks to the first two procedures are the same: the add-ons you select are installed and rolled out to the entire organization, but MATLAB users cannot install any custom add-ons.

Alternatively, you can choose the add-ons directory during user sign in, as long as users have write access to the directory. Using this procedure, MATLAB users can install any custom add-on**,** but you cannot roll out any add-ons to the users. The add-ons must be installed by the users themselves.

> **Note:** If the addons: *add-ons-directory* value is empty or not defined, *startDirectory*/MATLAB Add-Ons is the default directory for the add-ons.

From the following options, select the one that works best for your organization.

10.2.2.2.1 Option 1: Read-Only Add-Ons Directory Inside Docker Container

During this procedure, an add-ons directory is created inside the MATLAB installation directory and created as a Docker image. The add-ons you select are installed and rolled out to the entire organization, but MATLAB users cannot install any custom add-ons.

In matlab-pool.yaml (override), add the add-ons directory to the storage profile as shown:

```
storage:
  profiles:
    - name: nfsHome
      startDirectory: "/home/${subject.subjectId}"
      mounts:
        - name: home
          permissionType: user
          mountPath: "/home/${subject.subjectId}"
          uid: "${subject.uid}"
          gid: "${subject.gid}"
          type: nfs
          server: "nfs"
          subPath: "/exports/home/${subject.subjectId}"
          mountOptions:
"rw,relatime,vers=3,rsize=1048576,wsize=1048576,namlen=255,acregmi
n=600,acregmax=600,acdirmin=600,acdirmax=600,hard,nocto,noacl,prot
o=tcp,timeo=600,retrans=2,mountproto=tcp,local_lock=none,nolock"
      restrictNavigationOutsideMounts: false
      addons:
        directory: "/MATLAB/MATLAB Add-Ons"
```

10.2.2.2.2 Option 2: Read-Only Add-Ons Directory Through Special Mount

The add-ons you select are installed and rolled out to the entire organization. MATLAB users cannot install any custom add-ons.

1. Create a MATLAB Docker image. See Step 7. Prepare Docker Images in the MATLAB Online Server installation instructions.
2. Mount the add-ons directory through an external mount using Storage profiles.

In the matlab-pool.yaml (override), in the storage profile configuration, add the add-ons directory as shown:

```
storage:
  profiles:
    - name: nfsHome
      startDirectory: "/home/${subject.subjectId}"
      mounts:
        - name: home
          permissionType: user
          mountPath: "/home/${subject.subjectId}"
          uid: "${subject.uid}"
          gid: "${subject.gid}"
          type: nfs

          server: "nfs"

          subPath: "/exports/home/${subject.subjectId}"

          mountOptions:
"rw,relatime,vers=3,rsize=1048576,wsize=1048576,namlen=255,acregmi
n=600,acregmax=600,acdirmin=600,acdirmax=600,hard,nocto,noacl,prot
o=tcp,timeo=600,retrans=2,mountproto=tcp,local_lock=none,nolock"

       - name: addons

          mountPath: /AddOns

          type: nfs

          permissionType: none

          server: "nfs"

          subPath: "/exports/matlab/R2020a/addons"

          mountOptions:
"ro,relatime,vers=3,rsize=1048576,wsize=1048576,namlen=255,acregmi
n=600,acregmax=600,acdirmin=600,acdirmax=600,hard,nocto,noacl,prot
o=tcp,timeo=600,retrans=2,sec=sys,mountproto=tcp,local_lock=none,n
olock"      restrictNavigationOutsideMounts: false

      addons:

        directory: "/AddOns"
```

10.2.2.2.3 Option 3: Writable User Installed Add-Ons Directory

You choose the add-ons directory during user sign in. This directory must have write access for all users.

Using this procedure, MATLAB users can install any custom add-on, but you cannot roll out any add-ons to the users. The add-ons must be installed by the users themselves.

1. Create a MATLAB Docker image. See Step 7. Prepare Docker Images in the MATLAB Online Server installation instructions.
2. In the matlab-pool.yaml (override), in the storage profile configuration, add the add-ons directory as shown:

```
storage:

  profiles:

    - name: nfsHome
```

```
      startDirectory: "/home/${subject.subjectId}"

      mounts:

        - name: home

          permissionType: user

          mountPath: "/home/${subject.subjectId}"

          uid: "${subject.uid}"

          gid: "${subject.gid}"

          type: nfs

          server: "nfs"

          subPath: "/exports/home/${subject.subjectId}"

          mountOptions:
"rw,relatime,vers=3,rsize=1048576,wsize=1048576,namlen=255,a
cregmin=600,acregmax=600,acdirmin=600,acdirmax=600,hard,noct
o,noacl,proto=tcp,timeo=600,retrans=2,mountproto=tcp,local_l
ock=none,nolock"

restrictNavigationOutsideMounts: false

      addons:

        directory: "/home/${subject.subjectId}/MATLAB Add-
Ons
```

10.3 Update MATLAB Release

MATLAB Online Server supports MATLAB R2020a. MathWorks may release MATLAB updates following the R2020a general release (an update is an incremental update to the current release; an upgrade is a change from one major release to the next).

Use the following procedure to apply MATLAB release updates.

1. On a Linux machine, download and install the latest MATLAB R2020a update.
2. Create a backup of the working MATLAB image. This action might include pushing it to another repository or pushing it to the same repository but with a different tag.

> **Note:**
> If the MATLAB update is not successful, you need the backup image to restore MATLAB in MATLAB Online Server. Do not skip this step.

- You can use the following commands to save a backup of the existing MATLAB image:

```
docker tag <MATLAB_IMAGE_NAME>:<MATLAB_IMAGE_TAG>
<MATLAB_IMAGE_NAME>:<MATLAB_IMAGE_TAG_UPDATED>
```

For example:

```
docker tag
mathworks/mos/com.mathworks.matlabonlineserver.matlab-
image:R2020a
mathworks/mos/com.mathworks.matlabonlineserver.matlab-
image:R2020a-backup
```

- For a remote registry, run the following command also:

```
docker push <MATLAB_IMAGE_NAME>:<MATLAB_IMAGE_TAG_UPDATED>
```

For a remote registry, the two commands together might look similar to this example:

```
docker tag remote-
registry.com/mos/com.mathworks.matlabonlineserver.matlab-image:R2020a
remote-registry.com /mos/com.mathworks.matlabonlineserver.matlab-
image:R2020a-backup

docker push remote-registry.com
/mos/com.mathworks.matlabonlineserver.matlab-image:R2020a-backup
```

3. Use `mosadm` to build a new MATLAB Docker image. Run this command, replacing *MATLAB-install-location* with your actual MATLAB install location:

```
mosadm build-matlab-image MATLAB-install-location
```

This command overrides the existing MATLAB image.

4. Use `mosadm` to redeploy the MATLAB Pool pod using the following command:

```
mosadm undeploy matlab-pool
```

10.4 Configure Network Policies

10.4.1 Overview

A network policy specifies how groups of pods are allowed to communicate with each other and with other network endpoints. This section describes how to enable network policies and customize rules for providing communication with external systems, for example, LDAP, databases, and other internal services.

10.4.2 Prerequisites

Before you begin, make sure you have completed the following tasks:

- Install and configure MATLAB Online Server. See Single-Machine Installation.
- Familiarize yourself with Kubernetes network policies. See Network Policies at kubernetes.io.
- Create a Kubernetes cluster with a network policy plugin installed.
    - If you have a Kubernetes cluster that was created using mosadm bootstrap--node, Kube Router is preconfigured as a network policy plugin.
    - If you have a managed Kubernetes cluster (either from a cloud provider or a custom setup in-house Kubernetes cluster), you must install and validate the network policy adapter before following the instructions in this section. (If you need to, refer to Install a Network Policy Provider at kubernetes.io. Make sure you that the instructions you follow align with the Kubernetes version you are using).
    - If you are already using the network policy adapter, then you can proceed without any further setup.

10.4.3 Introduction

MATLAB Online Server has the following deployments running inside a Kubernetes cluster:

- AuthNZ (authentication)
- Core-UI (UI-related files)
- Gateway (proxies the requests to the appropriate resource)
- MATLAB Pool (remote MATLAB resources)
- Resource (acquires the MATLAB resource for the user)

The network policies are enabled for all the services and deployments by default. The following sections discuss how to disable, enable, and customize network policies, and how to know which service needs to be customized on a use-case basis.

10.4.4 Enable and Disable Network Policies

Network policies are enabled by default for all services and deployments.

- To enable or disable all network policies, make the following YAML configuration part of the respective microservice's override:

```
networkPolicy:
  enabled: true
```

- You can make the above configuration part of `all.yaml`.

10.4.5 Integrations

Integrations determine where you add the network policy override. Network policies must be customized based on the use case and the service or deployment that affect the policies.

10.4.5.1 LDAP

Lightweight Directory Access Protocol (LDAP) is an open, cross-platform protocol for accessing information from a server.

AunthNZ is one of the services in MATLAB Online Server that is responsible for communicating with the LDAP server to authenticate the user.

For LDAP integration, you must customize the authnz.yaml override file. See LDAP and Authorization.

10.4.5.2 Database

Database Toolbox™ provides functions and an app for exchanging data with relational and nonrelational databases. The toolbox enables this exchange by automatically converting between database and MATLAB® data types.

Database Toolbox (on Linux) supports any JDBC-compliant relational database, as well as NoSQL databases including Cassandra®, MongoDB®, and Neo4j®.

For instructions on connecting to a database using MATLAB, see the Database Toolbox documentation on the MathWorks website.

Databases expose a port to communicate with JDBC-compliant clients, such as MATLAB. For clients to communicate with the database server, you must customize the firewall and network policy in the matlab-pool.yaml override file to open the port to communicate.

10.4.5.3 External Web Services

The MATLAB® RESTful web services functions webread, websave, webwrite, and weboptions allow non-programmers to access many web services using HTTP GET and POST methods. However, some interactions with a web service are more complex and require functionality not supported by the RESTful web services functions. Use the HTTP interface classes for writing customized web access applications.

For more information on web access using MATLAB, see Web Access on the MathWorks website.

These web services also expose a port (typically 80 and 443) to communicate with clients, such as MATLAB. For clients to communicate with these services, you must customize the firewall and network policy in the matlab-pool.yaml override file to open the port to communicate. See MATLAB Pool.

10.4.6 Customize Network Policies

The network policies must be customized based on the use case and the service or deployment that affect the policies. See Integrations for details on which service policies must be customized.

Securing network connections revolves around defining egress and ingress rules. For a Kubernetes pod:

- Ingress refers to incoming connections to the pod.
- Egress refers to outgoing connections from the pod.

In Kubernetes network policy, you create "allow" rules for ingress and egress independently (egress, ingress, or both). MATLAB Online Server provides the ability to customize egress rules only through overrides. After you specify the egress in the overrides, the default policy is overridden.

Note:

If you have use cases to customize the ingress rules, contact MathWorks Support.

10.4.6.1 Override Egress Rules

To override egress rules, see the specification Network Policy v1 beta1 extensions on kubernetes.io.

An example of a network policy override defined in the MATLAB Online Server override YAML file follows.

```
networkPolicy:
  egress:
    - ports:
      # allow dns access
      - port: 53
```

Another example of a network policy override follows.

```
networkPolicy:
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: allow-egress-same-namespace
  namespace: default
spec:
  podSelector:
    matchLabels:
      color: blue
  egress:
      ports:
      # allow dns access
      - port: 53
```

You can see similar examples at Get Started with Kubernetes Network Policy at projectcalico.com.

10.4.6.1.1 Default Network Policy Egress Overrides for AuthNZ

The following entries in auth.yaml set the override for egress:

```
networkPolicy:
  egress:
    - ports:
      # allow dns access
      - port: 53
        protocol: TCP
      - port: 53
        protocol: UDP
      # allow standard http/https port access
      - port: 80
        protocol: TCP
      - port: 443
    protocol: TCP
   # allow standard ldap(s) port access
   - port: 389
    protocol: TCP
   - port: 636
    protocol: TCP
```

As specified by this policy override, the AuthNZ microservice can talk to the following ports:

- 53 – Used to make DNS lookups
- 80, 443 – Used to communicate with other HTTP(s) services (for example, http(s)://www.mathworks.com)
- 389, 636 – Used to communicate with the LDAP(s) server (for example, ldap:389)

10.4.6.1.2 Default Network Policy Egress Overrides for MATLAB Pool

The following settings in matlab-pool.yaml set the egress overrides:

```
networkPolicy:
 egress:
  - to:
   - ipBlock:
     cidr: 0.0.0.0/0
     except:
      - 169.254.169.0/24
   ports:
   # allow dns access
   - port: 53
    protocol: TCP
   - port: 53
    protocol: UDP
   # allow standard http/https port access
   - port: 80
    protocol: TCP
```

```
      - port: 443
        protocol: TCP
      # allow access to the flexlm server port
      - port: 27000
        protocol: TCP
      - port: 27001
        protocol: TCP
      # allow access on a variety of nfs related ports
      - port: 111
        protocol: UDP
      - port: 111
        protocol: TCP
      - port: 612
        protocol: UDP
      - port: 2049
        protocol: TCP
      - port: 2049
        protocol: UDP
      - port: 20048
        protocol: UDP
      - port: 20048
        protocol: TCP
      - port: 33668
        protocol: UDP
      - port: 33743
        protocol: TCP
      - port: 38494
        protocol: UDP
      - port: 36663
        protocol: TCP
      - port: 38494
        protocol: UDP
      - port: 40051
        protocol: TCP
      - port: 52241
        protocol: UDP
```

As specified in this policy override, MATLAB can talk to the following ports and IP addresses.

Ports:

- 53 – Used to make DNS lookups
- 80, 443 – Used to communicate with other HTTP(s) services (for example, http(s)://www.mathworks.com)
- 27000, 27001 – Used to communicate with the network license manager for checking out MATLAB licenses
    - These ports are controlled by license.dat.
    - See License Service Issues for information on how to make sure these ports are fixed in the license file.
- 111, 612, 2049, 20048, 33668, 36663, 33743, 38494, 40051, 52241 – Used for communicating with the NFS server
    - These are standard ports, and their actual values can vary based on the internal NFS configuration.
    - These ports can be derived leveraging the Linux commands portmap or lsof on the NFS server.

IP addresses:

- Allows access to all the IP addresses except the following range: 169.254.169.0/24
    - In this example, the IP range 169.254.169.0/24 is blocked to prohibit access to the EC2 metadata service (if the node is running on AWS).

10.4.7 Network Policy Override Examples

Determine where to add the network policy override and choose the appropriate override YAML file (such as authnz.yaml or matlab-pool.yaml). See Post-Installation Additional Configuration.

After consulting the information presented in this section, add the desired configuration and perform an upgrade (undeploy and deploy) of the service using mosadm (see Manage MATLAB Online Server).

10.4.7.1 Example: MATLAB Communication with MySQL Database

Database details:

- Server: mymsql.company.com

- JDBC port: 3306 (the standard one)
- Override file: matlab-pool.yaml

For MATLAB to communicate with the database server, the policy must allow access to the following ports:

- 53 – mwsql.company.com needs a DNS lookup
- 3306 – JDBC client (MATLAB) needs to communicate with the JDBC server (database)

The overall network policy configuration is specified as shown:

```
networkPolicy:
  egress:
    - to:
      - ipBlock:
          cidr: 0.0.0.0/0
          except:
            - 169.254.169.0/24
      ports:
      # allow dns access
      - port: 53
        protocol: TCP
      - port: 53
        protocol: UDP
      # allow standard http/https port access
      - port: 80
        protocol: TCP
      - port: 443
        protocol: TCP
      # allow access to the flexlm server port
      - port: 27000
        protocol: TCP
      - port: 27001
        protocol: TCP
      # allow access on a variety of nfs related ports
      - port: 111
        protocol: UDP
      - port: 111
        protocol: TCP
      - port: 612
        protocol: UDP
      - port: 2049
        protocol: TCP
      - port: 2049
        protocol: UDP
      - port: 20048
        protocol: UDP
      - port: 20048
        protocol: TCP
      - port: 33668
        protocol: UDP
      - port: 33743
        protocol: TCP
      - port: 38494
        protocol: UDP
      - port: 36663
        protocol: TCP
      - port: 38494
        protocol: UDP
      - port: 40051
        protocol: TCP
      - port: 52241
        protocol: TCP
      - port: 3306
  protocol: TCP
```

For more details, see Database Setup with MATLAB Online Server.

10.4.7.2 Example: Communication with Internal LDAP Server

LDAP server details:

- Server: myldap.company.com
- LDAP port: 343
- LDAPS port: 643
- Override file: authnz.yaml

For AuthNZ to communicate to the LDAP server, the policy must allow access to the following ports:

- 53 – myldap.company.com needs a DNS lookup
- 343, 643 – AuthNZ needs to bind to the LDAP(s) server

The overall network policy configuration is specified as shown:

```
networkPolicy:
  egress:
    - ports:
      # allow dns access
      - port: 53
        protocol: TCP
```

```
            - port: 53
              protocol: UDP
            # allow standard http/https port access
            - port: 80
              protocol: TCP
            - port: 443
              protocol: TCP
            # allow standard ldap(s) port access
            - port: 343
              protocol: TCP
            - port: 643

          protocol: TCP
```

See LDAP and Authorization for more details.

10.5 Log Rotation

Kubernetes does not currently rotate logs. You can configure log rotation manually. For example, you can deploy logrotate as a daemon service on a MATLAB Online Server node and configure it to run each hour.

MATLAB Online Server logs are under the folder /var/log/containers/. Use the logrotate config file to configure the logs. By default, this file is /etc/logrotate.conf. For more information, see Check Pod Status.

10.6 Update MATLAB Online Server

10.6.1 Overview

An update is a software change to the currently released version. MATLAB Online Server updates are released on a semi-regular basis to add functionality or provide bug fixes.

You must have MATLAB Online Server installed before you can install an update. See Single Machine Installation.

The method you use to update MATLAB Online Server creates two independent deployments (A and B) and then switches MATLAB Online traffic to the new deployment, resulting in little to no downtime for your end users.

10.6.2 Install and Deploy Update

To update MATLAB Online Server, you create a new server instance to deploy. Then you switch from the old instance (referred to here as deployment A) to the new instance (deployment B).

10.6.2.1 Step 1. Configure New Instance

1. Choose a deployment environment.

- If both the A and B deployments are on the same Kubernetes cluster, use a different Kubernetes namespace (NAMESPACE) for the new deployment. For example,
    - Existing deployment (A): **mathworksa**
    - New deployment (B): **mathworksb**
- If you are bootstrapping a single-node cluster (mosadm bootstrap-node), use a new server.
- If you are using a new Kubernetes cluster, configure the Kubernetes context on the terminal (KUBECONFIG config).

2. Choose a different domain name (DOMAIN_BASE) for the new deployment. For example:

- Existing deployment (A): **matlab.domain.com**
- New deployment (B): **matlab2.domain.com**

10.6.2.2 Step 2. Download MATLAB Online Server Update

1. Go to https://www.mathworks.com/products/matlab-online-server.html.
2. Download the MATLAB Online Server update file.
3. Extract the MATLAB Online Server update to a new folder.
4. Edit install.config for deployment B, using the same information you used for deployment A, except:

    - For DOMAIN_BASE, use the new domain (for example, matlab2.domain.com).

    - If you are using the same Kubernetes cluster, use the new NAMESPACE ((for example, mathworksb).

5. Deploy the MATLAB Online Server on the new server or new namespace, following the instructions for deployment (see Step 9. Deploy MATLAB Online Server).

   You must keep a couple of things in mind:

    - The number of namespaces cannot exceed the number of clusters.

    - Make sure you use the correct mosadm for the deployment you are working with. For example:

        - Deployment A: Use mosadm from the MATLAB Online Server A folder.

        - Deployment B: Use mosadm from the MATLAB Online Server B folder.

6. Verify that you have successfully installed the new MATLAB Online Server instance by navigating to the new DOMAIN_BASE (for example, matlab2.domain.com).

10.6.2.3 Step 3. Switch to Deployment B

1. In the B deployment, in the <OVERRIDES_DIR>/<cluster>/<namespace>/all.yaml file, update the domainBase value (matlab2.domain.com) to the original DOMAIN_BASE from deployment A (matlab.domain.com).

2. Make deployment B ready for DNS cutover by issuing the following command:
   mosadm upgrade --namespace mathworksb

3. Verify that you have successfully updated all pods; see Check Pod Status.

4. To test with a restricted set of users or on a single machine before switching the DNS to the new deployment, perform a local verification. See the note after these instructions.

5. Perform the DNS cutover to redirect the traffic to the new deployment (B) from the old deployment (A) using your vendor’s instructions to update DNS records. Depending on your DNS provider, switching the DNS might require taking the MATLAB Online server offline for some time.

After the DNS switch, all MATLAB users must log in again.

If you encounter any failures, see Problem Solving in MATLAB Online Server for further investigation. You can also start this procedure over.

Note: Local Verification

Optionally, before switching the DNS to the new deployment, you can test with a restricted set of users, or on a single machine. For example, update your operating system's hosts file to point your original domain base with the new server's IP address. To do so, update the hosts file:

- Linux: `/etc/hosts`
- Windows®: `<Windows Install>\System32\drivers\etc\hosts`
- macOS: `/etc/hosts`

Add the following line:

```
<new ip address>   <old DOMAIN_BASE>
```

For example:

```
52.201.253.135   matlab.domain.com
```

This update means that when DOMAIN_BASE is accessed from this machine, the requests go to deployment B. You can now verify the MATLAB Online Server installation for deployment B, using the verification process in Step 10. Verify MATLAB Online Server Installation of the MATLAB Online Server installation instructions.

10.7 MATLAB Online Server in an HA Environment

10.7.1 Overview

MATLAB Online Server is a collection of microservices that work together to provide users access to MATLAB through the web. This section describes how these individual components work in a highly available (HA) environment.

MATLAB Online Server has the following deployments running inside a Kubernetes cluster:

- AuthNZ – Authentication
- Core-UI – UI-related files
- Gateway – Proxies the requests to the appropriate resource
- MATLAB Pool – Remote MATLAB resources
- Resource – Acquires the MATLAB resource for the user

10.7.2 MATLAB Online Server Services Data

All the MATLAB Online Server services are stateless. They maintain state through cookies and communicating between each other.

All the data is stored in memory and not backed up. Therefore, in case of failure or restart of the pod, MATLAB users might need to sign in again.

10.7.3 MATLAB Data Persistence

MATLAB users write, store, and execute their code and data on MATLAB Online Server. You can configure multiple storage options: some are persistent between user sessions and some are nonpersistent (ephemeral).

Depending on your configuration of storage profiles, additional training for the MATLAB user might be required. If the MATLAB user saves data in the persistent folders or mounts, when the user signs into a new session, all saved data is restored. However, data that the user stores in a nonpersistent folder, such as /tmp, does not persist to the next session. Make sure your users are familiar with the concept of data persistence and know which folders to use for data they want to preserve between sessions.

For information on mounting persistent folders such as NFS mounting or host mounting, see Storage Profiles.

10.7.4 High Availability

Because MATLAB Online Server services are stateless, they communicate with each other for maintaining state. Data such as login information and the mapping between the user session and the remote MATLAB is stored in memory. This data is also not persistent. For this reason:

- If the number of replicated deployments is more than 1, as the data is stored in memory, the load balancer/ingress controller might send the requests to a replica that does not have the correct data. This can cause the session to go to multiple logins.
- If the service is restarted, the user might need to sign in again.

Part 3: End User Guides

11 End User Guide to MATLAB Online

11.1 MATLAB Online Using MATLAB Online Server

11.1.1 Introduction

MATLAB Online Server provides access to MATLAB Online from any standard web browser, is centrally hosted and managed by your organization, and integrates with your network file system and authentication services.

This topic explains the key features of MATLAB Online and some of the important variations from desktop MATLAB.

11.1.2 How to Access MATLAB Online

In the browser, enter the URL provided by your administrator. Sign in to MATLAB Online using your organization's credentials. If you have questions about the sign-in process, contact your administrator.

If your administrator has set up multiple MATLAB versions, after you sign on, you can select from the available options.

11.1.3 Files

MATLAB Online integrates with your network file system as defined by your administrator. The Current Folder browser displays the root of this file system. If you have questions about the Current Folder browser, contact your administrator.

File Operation	Description
Current Folder browser	Use the Current Folder browser to interactively manage files and folders on MATLAB Online. With the Current Folder browser, you can view, create, open, move, and rename files and folders in the current folder.
File Path Information	Because MATLAB Online Server runs on a Linux platform, the file and path separator characters you use in MATLAB Online are different from those in desktop MATLAB on a Microsoft Windows platform. The following commands can help you adjust your code to a Linux platform: `fullfile` builds full file names from component parts with the appropriate file separator characters. `filesep` returns the platform specific file separator character. `fileparts` gets parts of file name. `path` lets you view or change search path. `pathsep` searches path separator for current platform.
File Management	• Upload

	To upload files within MATLAB Online, use the **Upload** button on the toolstrip or drag and drop your file into the Current Folder browser. • Download To download a file, use the toolstrip button or the command line. You must have access to the file.
MEX Command	The MEX command, for building C/C++ or Fortran MEX files, is not supported.

11.1.4 System

Specification	Value
Operating system	Linux
MATLAB version	R2020a
Compute resources	Check with your administrator for exact resources. The recommendation is a minimum of 4 cores and 8 GBs per user.
Support browsers	This product is compatible with most modern web browsers running on Windows, Mac, Linux, and Chrome OS, specifically, the current versions of Mozilla Firefox, Apple Safari, and Microsoft Edge. For the best overall experience, Google Chrome is recommended.

11.1.5 MATLAB Sessions

Operation	Description
Session Timeout	The default idle time for MATLAB Online is 15 minutes. Idle time is determined by inactivity within the user interface. Your administrator can choose to increase the idle time duration.
Session Save and Restore	The following items are automatically saved and restored between MATLAB Online sessions: • MATLAB path • MATLAB settings and preferences • Files in your home folder The following items are not automatically saved and restored between sessions: • Workspace variables

	• Dynamic Java classpath

11.1.6 Add-Ons and Supported Toolboxes

By default, add-ons, support packages, and third-party toolboxes are controlled and installed by your administrator. Administrators can also configure MATLAB Online Server to allow end users to install MATLAB toolbox add-ons (.mltbx). Your administrator can review Install Support Packages for MATLAB Online and Install Add-Ons for MATLAB Online for more information.

The following table lists MathWorks toolboxes that are supported in MATLAB Online running on MATLAB Online Server. Check with your administrator for information on specific licenses.

Toolbox	Exception
Aerospace Toolbox	• aeroDataPackage • Aero.FGAnimation • astfganim • Aero.VirtualRealityAnimation
Communications Toolbox	RF propagation functionality
Computer Vision Toolbox	• vision.VideoPlayer System object • vision.DeployableVideoPlayer System object • Camera Calibrator app • Stereo Camera Calibrator app • Training Image Labeler app
Control System Toolbox	
DSP System Toolbox	• dsp.AudioFileReader • dsp.AudioFileWriter • audioDeviceWriter • C/C++ and MEX code generation from MATLAB algorithms in DSP System Toolbox • Simulink blocks in the DSP System Toolbox library • Filter Builder app
Deep Learning Toolbox	
Econometrics Toolbox	
Financial Instruments Toolbox	
Financial Toolbox	
Fuzzy Logic Toolbox	
Global Optimization Toolbox	
GPU Coder	• GPU Coder app • Configuration parameter dialog box • coder.screener

	• GPU Coder Support Package for NVIDIA GPUs • Deep Learning code generation for TensorRT targets
Image Processing Toolbox	• implay • immovie • Image Segmenter • Image Region Analyzer app • Image Batch Processor app
Mapping Toolbox	
MATLAB Coder	• MATLAB Coder app • Configuration parameter dialog box • coder.screener • MATLAB Coder Interface for Deep Learning Libraries Support Package • MATLAB Coder Support for Raspberry Pi • Deep Learning code generation for MKL-DNN targets
Model Predictive Control Toolbox	
Optimization Toolbox	
Partial Differential Equation Toolbox	
Phased Array System Toolbox	
RF Toolbox	
Robust Control Toolbox	
Signal Processing Toolbox	Filter Builder app
Statistics and Machine Learning Toolbox	
Symbolic Math Toolbox	MuPAD Notebook and functions that interface with it
Wavelet Toolbox	• Wavelet Analyzer app • Wavelet Signal Denoiser app • Signal Multiresolution Analyzer app

11.1.7 Functionality

11.1.7.1 Keyboard Shortcuts

Action	Keyboard Shortcut
Move forward through the toolstrip and other tools.	**Ctrl+F6** On macOS systems, use **Cmd+F6** instead.
Move backward through the toolstrip and other tools.	**Ctrl+Shift+F6**

	On macOS systems, use **Cmd+Shift+F6** instead.
Move into a tool.	**Tab**
Move between controls within a tool.	**Up/Down Arrow**, **Left/Right Arrow**
Close controls within a tool, for example, documents in the Editor and Live Editor.	**Delete**
Open context menu on Windows® and UNIX® systems.	**Shift+F10**
Show access keys for the toolstrip.	**Alt**
Open a toolstrip tab and show access keys for the toolstrip.	**Alt+<*character*>** For example, pressing **Alt** followed by **H** accesses the **Home** tab and displays access keys for the features available on that tab.
Display a compact list of keyboard shortcuts.	**Ctrl+/**

11.1.8 Window Management

- Saving workspace variables using **Ctrl+S** is not supported.
- Because MATLAB Online runs in a web browser, most windows that MATLAB opens are constrained to the browser window.
- Some apps open in new browser tabs.

11.1.9 Connecting Hardware

MATLAB Online cannot interact with most other hardware, including instrument control.

However, the following exceptions apply:

- It can interact with USB webcams only through Google Chrome.
- It can communicate with Raspberry Pi hardware.

11.1.10 Other Limitations

- Packaging tools for add-ons and MATLAB Compiler and MATLAB Compiler SDK are not supported.

- Windows-specific components, such as COM, are not supported.
- xlsread and xlswrite work in basic mode.
- The graphical interface to the profiler is not supported.
- Use of the shell escape bang (!) command is not fully supported.

11.2Database Access with MATLAB Online

11.2.1 Database Setup with MATLAB Online

This topic demonstrates how to leverage database drivers and a set of configurations to query your desired database from MATLAB Online.

1. The database you want to connect to might have already been enabled for your installation. To find out, contact your administrator. MATLAB Online, after being installed, requires additional configuration to allow it to communicate with your company databases.

2. Upload your database driver. Common JDBC Drivers are available from the Database Toolbox Driver Installation page on the MathWorks website.
   a. Create a folder named `MyDatabaseDrivers`.
   b. Upload the JAR or zip file containing your database driver.
   c. Verify your folder and file are on the MATLAB Online path by right-clicking on the folder, selecting **Add to Path**, and choosing **Selected Folders and Subfolders**. See What Is the MATLAB Search Path? on the MathWorks website, which gives an overview of the search path for desktop MATLAB.

3. Learn how to query a database using MATLAB. Search "database" in the Help Center for more details, or see the Database Toolbox documentation on the MathWorks website.

The following example m code for MySQL can help you get started:

```
% Configure the database vendor type
     opts = configureJDBCDataSource("Vendor", "MySQL
% Setup your data base connection options, replacing the < > values
with your
% desired settings
% <MySQL-Data-Base> is a user friendly name for your Data Basesetup
% <database.mycompany.com> is the server name of the database
% <portnumber> is the port number of the database
% </path/to/driver.jar> is the location your uploaded your driver to
%     For example:
%     /home/username/mysql-connector-java-8.0.18.jar
opts = setConnectionOptions(opts, "DataSourceName", "<MySQL-
Database-Name>","Server",
"<database.mycompany.com>", "PortNumber", <portnumber>,
"JDBCDriverLocation", "</path/to/driver.jar>");

% Configure with your database’s username and password
username='<database_username>';
password='<database_password>';

% testConnection will validate the connectivity from your MATLAB
session to the
```

```
% Database. If status = 1, the connection was successful.
[status, message] = testConnection(opts, username, password);

% Saves your JDBC configuration as your <MySQL-Data-Base>
saveAsJDBCDataSource(opts);

% Establish a connection to your database
    conn = database('<MySQL-Data-Base>', username, password);
% Construct your SQL statement for your schema’s <database.table>
statement = 'select * from <database.table> limit 10';
data = select(conn, statement);
```

If you encounter any difficulty querying the database, try the following tips to resolve the issue:

- Review Troubleshooting in Database Toolbox on the MathWorks website.
- If you encounter any of the errors in the following table, apply the suggested solution.

Error	Suggested solution
requires Database Toolbox	While running any commands, if you receive this error message, contact your administrator and request that they enable Database Toolbox for your environment.
JDBC data source does not contain driver location	Validate that your driver is available on your MATLAB Online path.
JDBC Driver Error: Communications link failure	Check with your administrator that the database server and port have been configured for MATLAB Online Server.

11.3 Set Up GitHub for Use with MATLAB Online Server

11.3.1 GitHub Setup

You can interact with GitHub repositories from your MATLAB Online Server command window. This section contains examples that work with other Git repositories; simply substitute the correct server name in the command prompt examples.

If your IT administrator has configured a home directory on a network drive that can be shared between your computer and MATLAB, you can leverage any Git tool of your choice from your computer. The directories and files you interact with in this shared directory are shown in the MATLAB folder browser.

Note:

- MATLAB Online Server does not currently offer support for user-interface-based source control.
- MATLAB Online Server does not have an interactive shell so it does not allow command line prompts for a Git user name or password.

11.3.2 Clone Public Repository

To clone a public repository, use this example, replacing the name of the repository with the one you want to clone:

```
!git clone https://github.com/kelseyhightower/nocode.git
```

or

```
system('git clone https://github.com/kelseyhightower/nocode.git')
```

11.3.3 Clone Private Repository

To clone a private repository from within MATLAB Online Server, provide your user name and password to the repository. Use the following example, providing your specific details where indicated in italics:

```
!git clone https://GitUser:Password@github.com/GitUser/repo.git
```

11.4 Parallel Computing Toolbox and MATLAB Distributed Computing Server—End Users

MATLAB Online Server includes support for Parallel Computing Toolbox and the use of MATLAB Distributed Computing Server Clusters (also known as MATLAB Parallel Server in some documentation) from the MATLAB Online Server user interface command window.

You can access Parallel Computing Toolbox's parpool functionality from the command window. The different ways of using this functionality are illustrated in the following sections.

11.4.1.1 Use Local parpool Workers by Default

When you use a function like parfor or spmd, MATLAB Online Server automatically creates a parpool of local workers to run your code.

For example:

```
>> n = 100; A = 200; a = zeros(1,n);
>> parfor i = 1:n
     a(i) = max(abs(eig(rand(A))));
   end

Starting parallel pool (parpool) using the 'local' profile ...

Connected to the parallel pool (number of workers: 2).

>>
```

This behavior is exactly like desktop MATLAB, except that the number of local workers available to each MATLAB Online session is, by default, configured to 2 and can be configured by the MATLAB Online Server Administrator.

11.4.1.2 Use MATLAB Distributed Computing Server Cluster

To use a MATLAB Distributed Computing Server Cluster, the MATLAB Online Server user must upload the profile for the cluster before using it. This profile is created by the MATLAB Distributed Computing Server cluster administrator or the desktop MATLAB user who creates an MJS-based cluster for their own use.

In MATLAB Online Server, the user uploads the cluster definition, imports it into MATLAB, and uses the parpool function to select that profile for use.

```
>> parallel.importProfile('MJSProfile1')

ans =

 'MJSProfile1'


>> parallel.clusterProfiles % displays the cluster profiles available for use

ans =

1x2 cell array

{'local'} {'MJSProfile1'}
>> parpool('MJSProfile1') % use the MJSProfile1 cluster profile to start a parpool

Starting parallel pool (parpool) using the 'MJSProfile1' profile ...

Connected to the parallel pool (number of workers: 4).
```

```
ans =

ClusterPool with properties:

Connected: true

NumWorkers: 4

Cluster: MJSProfile1

AttachedFiles: {}

AutoAddClientPath: true

IdleTimeout: 30 minutes (30 minutes remaining)

SpmdEnabled: true

EnvironmentVariables: {}
>>
```

To create an MJS cluster, see the following web pages:

- Discover Clusters and Use Cluster Profiles
- Set Up Resources

Also see Get Started with Parallel Computing Toolbox for more information about Parallel Computing Toolbox and MATLAB Distributed Computing Server.

11.4.1.3 Parallel Computing Toolbox Disabled by MATLAB Online Server Administrator

If your MATLAB Online Server administrator has disabled the use of Parallel Computing Toolbox or local workers, you can expect to see the following results in the command window:

```
>> parallel.clusterProfiles

ans =

1x1 cell array

{'local'}
>> parpool

Unable to run the 'parpool' function, because it is not supported for MATLAB Online. To

use this function, use an installed version of MATLAB on your computer.
>>
```

Part 4: Supporting Resources

12 Resources and Links

12.1 AuthNZ

Authentication and authorization. See https://en.wikipedia.org/wiki/Authentication.

12.2 Docker

https://www.docker.com/

12.3 FlexNet Publisher (FLEXlm)

https://www.flexerasoftware.com/monetize/products/flexnet-licensing.html

12.4 Helm Chart

https://www.coveros.com/what-is-a-helm-chart-a-beginners-guide/

12.5 Kubernetes

https://kubernetes.io/

12.6 NGINX

https://en.wikipedia.org/wiki/Nginx

12.7 Transport Layer Security (TLS)

https://en.wikipedia.org/wiki/Transport_Layer_Security

12.8 YAML

https://yaml.org/