

MATLAB® Online Server™

Installation and Administration Guide



MATLAB®

R2022b



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™ Installation and Administration Guide

© COPYRIGHT 2020-2022 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

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	Online only	New for Version 1.0 (R2020a)
April 2020	Online only	Revised for R2020a Update 1
July 2020	Online only	Revised for R2020a Update 4
September 2020	Online only	Revised for Version 1.1 (R2020b)
November 2020	Online only	Revised for R2020b Update 2
March 2021	Online only	Revised for Version 1.2 (R2021a)
June 2021	Online only	Revised for R2021a Update 3
September 2021	Online only	Revised for Version 1.3 (R2021b)
March 2022	Online only	Revised for Version 1.4 (R2022a)
August 2022	Online only	Revised for R2022a Update 5
September 2022	Online only	Revised for Version 1.5 (R2022b)

1 | **MATLAB Online Server Introduction**

Introduction to MATLAB Online Server	1-2
Purpose	1-2
Architecture	1-2
MATLAB Online Server Components	1-3
Requirements	1-6
Intended Audience	1-6
Supported Operating Systems	1-6
Software Requirements	1-6
System Requirements	1-6
License Requirements	1-7

2 | **MATLAB Online Server Installation**

Online Install on Single Machine	2-2
Verify Installation Prerequisites	2-2
Install Docker	2-2
Install Network License Manager	2-2
Download and Install MATLAB	2-3
Download MATLAB Online Server Installer	2-3
Install Kubernetes	2-3
Configure Server	2-4
Prepare Docker Images	2-7
Install NGINX Ingress Controller	2-7
Prepare Helm Charts and Configure Overrides	2-8
Deploy MATLAB Online Server	2-8
Verify MATLAB Online Server Installation	2-9
Next Steps	2-9
Offline Install on Single Machine	2-11
Verify Installation Prerequisites	2-11
Install Network License Manager	2-11
Download MATLAB for Offline Installation	2-12
Download MATLAB Online Server Installer	2-12
Build Offline Installer	2-12
Install MATLAB Online Server on Air-Gapped Machine	2-13
Next Steps	2-13
Install on Cloud-Managed Kubernetes	2-14
Verify Installation Prerequisites	2-14

Install Network License Manager	2-14
Download and Install MATLAB	2-15
Download MATLAB Online Server Installer	2-15
Configure Server	2-16
Prepare and Push Docker Images	2-19
Install NGINX Ingress Controller	2-19
Prepare Helm Charts and Configure Overrides	2-20
Deploy MATLAB Online Server	2-20
Connect to MATLAB Online Server	2-21
Verify MATLAB Online Server Installation	2-21
Next Steps	2-22
Install on Red Hat OpenShift	2-23
Verify Installation Prerequisites	2-23
Install Network License Manager	2-23
Download and Install MATLAB	2-24
Download MATLAB Online Server Installer	2-24
Configure Server	2-24
Prepare and Push Docker Images	2-27
Prepare Helm Charts and Configure Overrides	2-28
Deploy MATLAB Online Server	2-29
Connect to MATLAB Online	2-30
Verify MATLAB Online Server Installation	2-30
Next Steps	2-31
Installation Configuration Options for MATLAB Online Server	2-32
Domain Name	2-32
Docker	2-32
Namespace	2-33
Authentication	2-34
License Server	2-35
Security	2-35
Kubernetes	2-36
Base Container Image	2-36
Offline Installation	2-36
Client-Only Mode	2-36
Logging	2-37
High Availability	2-37
Dashboard Deployment	2-37
Firewall	2-38

Post-Installation Additional Configuration

3

Post-Installation Additional Configuration	3-2
Customize Organization Name and Logo	3-3
Specify Organization Name	3-3
Specify Logo	3-3
Implement Customized Sign-In Page	3-4

Authentication	3-5
Identity Provider	3-5
Local	3-6
LDAP	3-6
SAML	3-11
Configure MATLAB Pool	3-14
License Server	3-14
MATLAB Pods Ready at Login	3-14
MATLAB Installation Source	3-15
Hardware Resources	3-18
Storage	3-19
Authorization	3-38
Session Timeouts	3-38
MATLAB Pod Metadata and Environment Variables	3-39
Title Bar	3-40
Configure Multiple MATLAB Pools	3-41
Overview	3-41
Enable Multiple Pool Chooser	3-41
Create Multiple Pools	3-42
Configure Additional Pool Options	3-43
Specify MATLAB Documentation	3-43
Redeploy MATLAB Online Server	3-43
Add New Pool to MATLAB Pools Already Running	3-44

MATLAB Online Server Administration

4

Manage MATLAB Online Server	4-2
Manage Docker Images	4-2
Build MATLAB Image	4-4
Deploy, Upgrade, and Undeploy Services	4-4
System Information	4-8
Bootstrap or Reset Single-Node Kubernetes Cluster	4-9
Bootstrap Node	4-9
Reset Node	4-13
Configure Overrides	4-15
Kubernetes Configuration	4-15
Copy Helm Charts	4-15
Generate Overrides	4-15
Additional Configuration Options	4-16

5

Check Pod Status	5-2
Get Pod List	5-2
Get List of Other Required Services	5-3
Get Pod Descriptions	5-3
Syntax	5-3
Example Output	5-4
Check Logs	5-6
Access Logs Using the File System	5-6
Dump Logs to System Output	5-6
Stream Logs to System Output	5-7
Set Up Kubernetes Dashboard for MATLAB Online Server	5-8
Prerequisites	5-8
Install and Set Up Dashboard on Single-Node Configuration	5-8
Install and Set Up Dashboard on Managed Kubernetes Provider	5-10
Log Shipping	5-11
Option 1: Persistence of Logs on Host	5-11
Option 2: Integrate with Splunk (Advanced)	5-11
Problem Solving in MATLAB Online Server	5-14
MATLAB Pod Issues	5-14
Pod Creation Stuck	5-17
Evicted or Terminated Pods	5-20
Out-of-Resource Conditions	5-21
License Service Issues	5-23
Fix Security Exception Error When Accessing Help	5-26
Problem	5-26
Solution	5-26

Server Maintenance

6

Smoke Testing MATLAB Online Server	6-2
Prerequisites	6-2
Arguments for Smoke Test Option	6-2
Addressing Failures	6-5
Parallel Computing Toolbox and MATLAB Parallel Server - Administrators	6-9
Overview	6-9
Configure Support in MATLAB Pool	6-9
Enable Use of On-Premise MJS or HPC Clusters	6-10
End User Instructions	6-11

Database Setup with MATLAB Online Server	6-12
Basic Configuration	6-12
Advanced Configuration	6-12
End User Instructions	6-13

Product Maintenance

7

Install Support Packages for MATLAB Online	7-2
Prerequisites	7-2
Install Procedure	7-2
Install Add-Ons for MATLAB Online	7-4
Prerequisites	7-4
Install Procedure	7-4
Update MATLAB Release	7-8
Configure Network Policies	7-9
Overview	7-9
Prerequisites	7-9
Introduction	7-9
Enable and Disable Network Policies	7-9
Integrations	7-10
Customize Network Policies	7-11
Network Policy Override Examples	7-14
Log Rotation	7-17
Update MATLAB Online Server	7-18
Overview	7-18
Install and Deploy Update	7-18
MATLAB Online Server in an HA Environment	7-21
Overview	7-21
MATLAB Online Server Services Data	7-21
MATLAB Data Persistence	7-21
High Availability	7-21

MATLAB Online Server End User Guides

8

MATLAB Online Using MATLAB Online Server	8-2
Introduction	8-2
How to Access MATLAB Online	8-2
Files	8-2
System	8-3
MATLAB Sessions	8-4
Add-Ons and Supported Toolboxes	8-4

Functionality	8-6
Window Management	8-7
Connecting Hardware	8-7
Other Limitations	8-7
Database Access with MATLAB Online	8-8
Database Setup with MATLAB Online	8-8
Set Up GitHub for Use with MATLAB Online Server	8-10
GitHub Setup	8-10
Clone Public Repository	8-10
Clone Private Repository	8-10
Parallel Computing Toolbox and MATLAB Parallel Server—End Users	8-11
Use Local parpool Workers by Default	8-11
Use MATLAB Parallel Server Cluster	8-11
Parallel Computing Toolbox Disabled by MATLAB Online Server Administration	8-12

MATLAB Online Server Supporting Resources

9

Resources and Links	9-2
AuthNZ	9-2
Docker	9-2
FlexNet Publisher (formerly FLEXlm)	9-2
Helm Chart	9-2
Kubernetes	9-2
NGINX	9-2
PEM	9-2
SAML	9-2
Transport Layer Security (TLS)	9-2
YAML	9-2

MATLAB Online Server Introduction

Introduction to MATLAB Online Server

Purpose

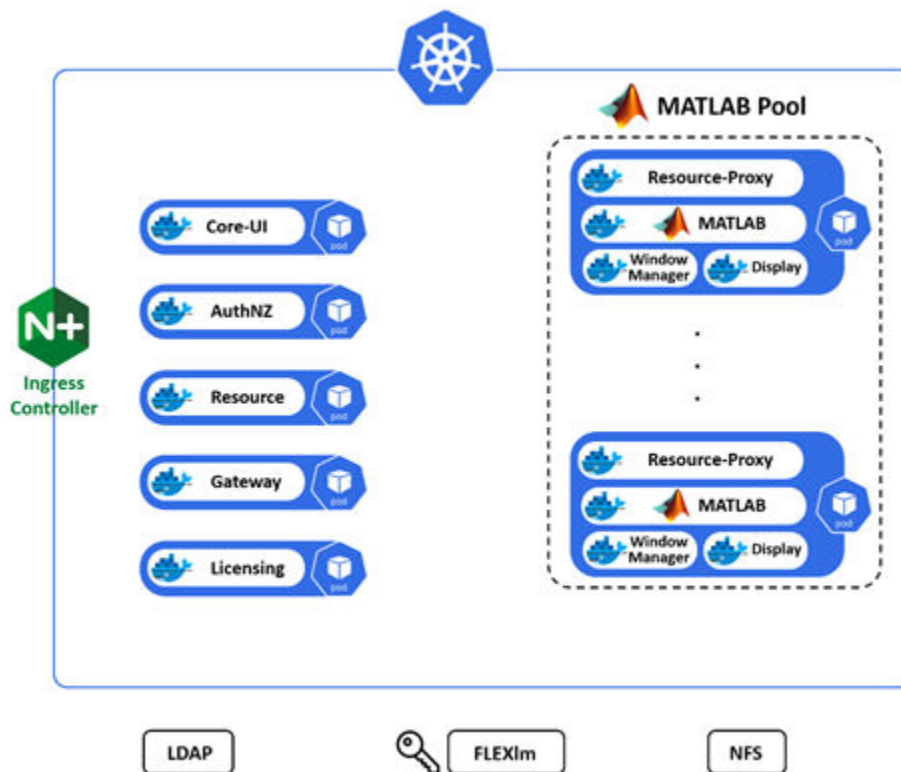
MATLAB Online Server lets you centrally host and manage MATLAB Online™ for the MATLAB users in your organization, enabling them to access MATLAB from their web browsers. You can integrate MATLAB Online Server with your network file system and authentication protocol, configure hardware resources, and 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 “Resources and Links” on page 9-2 for a list of many of the terms and third-party applications used when setting up MATLAB Online Server.

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.



MATLAB Online Server Components

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

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 then 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

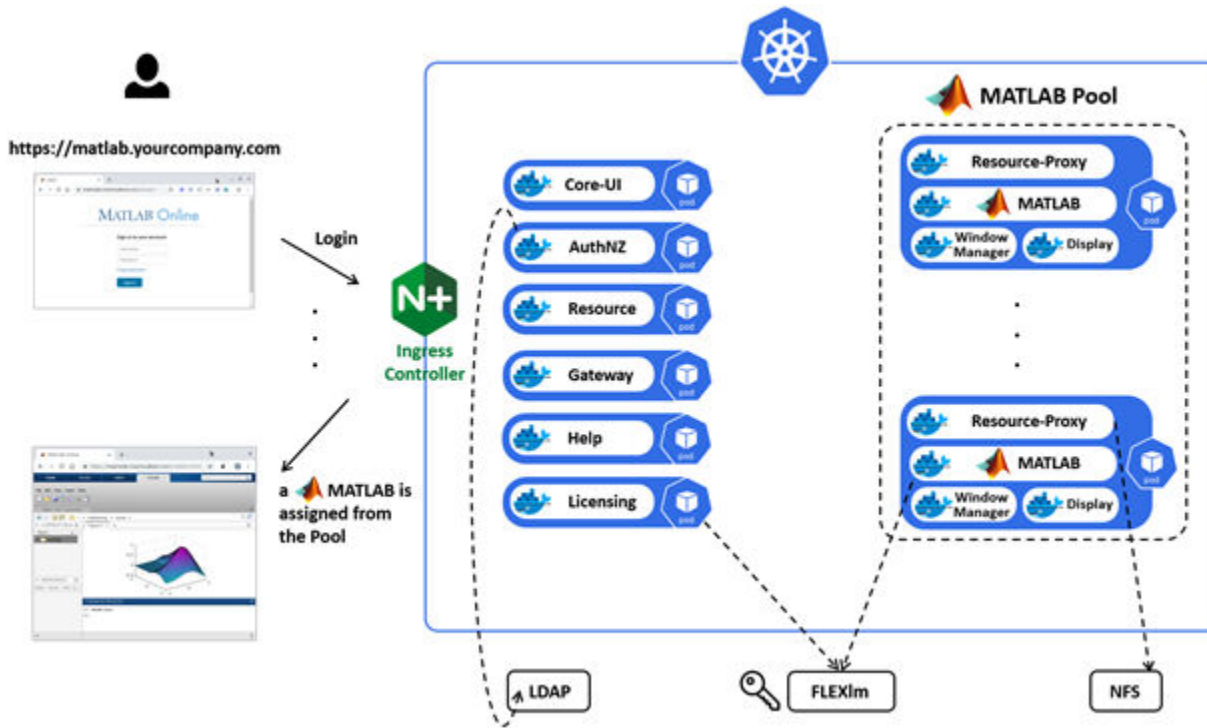
The Licensing component communicates with the MathWorks® License Manager, which uses [FlexNet Publisher](#) (formerly FLEXlm®) to check for MATLAB Online Server licenses.

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.

Workflow

The diagram below illustrates a high-level workflow for when a user signs into 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.

YAML and SAML

Configuration settings for all components reside in YAML files (security assertions in SAML). YAML and SAML file names must be all lower case and their contents must be strictly formatted. See [YAML 1.2](#).

To apply MATLAB Online Server service-specific configuration (such as AuthNZ, MATLAB-Pool, Core-UI), edit the settings in the appropriate YAML file, named and located in the following format:

```
<matlab_online_server_install_dir>/overrides/<cluster>/<namespace>/<service-name>.yaml
```

For example, `/opt/matlab_online_server/overrides/matlab-online-server/mathworks/authnz.yaml`.

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

Requirements

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 or other authentication protocols, and configuring networking settings including IP addresses and DNS entries, as well as experience with observing operational systems using logs and monitoring systems.

Supported Operating Systems

MATLAB Online Server is compatible with Linux only.

Supported operating systems:

- Ubuntu® 18.04 LTS, 20.04 LTS
- Red Hat® 7.7-7.9
- CentOS 7.7-7.8

Software Requirements

Kubernetes

The table shows the Kubernetes® versions supported for each MATLAB Online Server release.

MATLAB Online Server Version	Kubernetes Versions Supported
R2020a	1.15
R2020b	1.15-1.17
R2021a	1.17-1.18
R2021b	1.17-1.20
R2022a	1.18-1.23
R2022b	1.18-1.23

For instructions on installing Kubernetes, see “Install Kubernetes” on page 2-3.

Docker

Minimum supported version: Docker® 20

For instructions on installing Docker, see “Install Docker” on page 2-2.

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	120 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 four 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.

License Requirements

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.

MATLAB Online

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

MATLAB Online Server Installation

Online Install on Single Machine

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](#) in the Kubernetes documentation.

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 “Resources and Links” on page 9-2 for an introduction to the terms and processes used when installing MATLAB Online Server.

Note These instructions include `sudo` in all commands. Depending on how your environment is configured, `sudo` administration privileges might not be required for all commands.

Verify Installation Prerequisites

Before beginning the installation, verify the following:

- Your system is running one of the supported operating systems and meets the minimum system requirements described in “Requirements” on page 1-6.
- Your operating system is hosted at a static IP address. Kubernetes requires a static IP address to function properly. With dynamic IP addresses, when the IP address changes, the Kubernetes server fails and must be reinstalled.

Install Docker

Install the latest version of Docker. MATLAB Online Server, such as by using `sudo apt-get install docker.io`.

For instructions on installing Docker for your Linux distribution, see [Install Docker Engine](#) in the Docker documentation.

Install Network License Manager

Follow the Linux instructions for installing on an offline machine described in “Install License Manager on License Server” in the Help Center. 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.
- 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 will run without any errors and then will check out the license when it starts. If the license is invalid or the license server cannot be reached, then the license service will exit and then Kubernetes will automatically attempt 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 will then reject requests with a failed status. The logs of the services indicate when they enter or exit this hibernation or grace period state.

Download and Install MATLAB

Download and install the version of MATLAB that your MATLAB Online users will connect to.

Follow the Linux instructions in “Install Products Using Internet Connection” (in the Help Center) to install MATLAB (R2020a or later) on the server.

If your machine is not connected to the Internet, follow the instructions in “Install Products Using File Installation Key”.

Make sure you install MATLAB into the folder `/MATLAB`.

Download MATLAB Online Server Installer

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

- 1 Go to the [Downloads](#) page on mathworks.com.
- 2 Under **Select Release**, select the current release, for example, R2022b.
- 3 Expand the **Get MATLAB Online Server** section. This section appears only if your account has a license for MATLAB Online Server software.
- 4 Click **Download** to download the ZIP file.
- 5 In the download folder or folder of your choice, extract the installation files. For example, this command downloads the ZIP file to your home directory (`~/`).

```
unzip R2022b_matlab_online_server.zip -d ~/
```

The unzipped contents are as follows:

- `matlab_online_server/`
 - `data/`
 - `attributions.txt`
 - `install.config`
 - `thirdpartylicenses.txt`
 - `mosadm`

The `mosadm` command is the main utility to perform the remaining installation steps. For details on the various operations it can perform, run this command:

```
./mosadm help
```

Install Kubernetes

Download and install the required Kubernetes components.

- 1 Navigate to the extracted folder (for example, `~/matlab_online_server`) and execute this command:

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

This command downloads and then 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” on page 4-9 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 --kubeconfig /etc/kubernetes/admin.conf 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

- 5 Merge the `kube-config` file into your home folder so that you can execute the `kubectl` command without having to specify the `--kubeconfig /etc/kubernetes/admin.conf` flag going forward.

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

Configure Server

Edit the `install.config` file, located in the folder where you unzipped the installer. Before editing this file, consider creating a backup of `install.config` so that you have a record of the default settings.

In the file, modify these parameters:

Machine Configuration

Parameter	Description	Default
DOMAIN_BASE	<p>Fully qualified domain name used to access MATLAB Online Server from a web browser. For example: <code>matlabonline.mycompany.com</code>. Specifying an IP address as the domain base is not supported.</p> <p>It is recommended that the domain name is registered to your organization and that the IP address of the server or load balancer are available on the DNS server of the organization.</p> <p>To test that your domain is working, it is possible to temporarily update the host entry of your client machine to have the IP address and the fully qualified domain name. Then, you can test accessing MATLAB Online Server instance from a browser in your client using this domain name.</p>	<code>matlab.domain.com</code>

Authentication Configuration

Parameter	Description	Default
ML_PASSWORD	<p>MATLAB Online sign-in password.</p> <p>The user name for this bootstrapping mode is <code>admin</code>.</p> <p>For security purposes, it is recommended that you replace the default administration password with a more secure one.</p> <p>Identity provider configuration for authenticating MATLAB Online users is covered in "Authentication" on page 3-5.</p>	ML_PASSWORD=password

License Configuration

Parameter	Description	Default
MOS_LICENSE_SERVER	<p>MATLAB Online Server license server. Specify the port and hostname IP address in the format <code>port@host</code>.</p> <hr/> <p>Note If you have an Enterprise License, to correctly track usage, MathWorks requires two license servers: one for MATLAB Online Server (<code>MOS_LICENSE_SERVER</code> parameter) and one for MATLAB (<code>MATLAB_LICENSE_SERVER</code> parameter).</p> <p>With the two-server configuration, you must also have multiple pools (at least 2). See "Configure Multiple MATLAB Pools" on page 3-41.</p> <hr/> <p>Example: 14650@192.166.248.2</p>	"27000@flexlm"

Parameter	Description	Default
MATLAB_LICENSE_SERVER	MATLAB license server details. Specify the port and hostname IP address in the format port@host. Example: 27000@172.0.0.1	"27000@flexlm"

Additional Configuration Options

You can optionally modify additional parameters based on your configuration. For a complete list of parameters, see "Installation Configuration Options for MATLAB Online Server" on page 2-32.

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
```

The command loads the images from the data folder of your server into the local Docker cache. The program prompts you to verify that you have enough disk space to build and load the MATLAB image. To check the size of your /MATLAB folder, you can use the du (disk usage) command. For example:

```
du ./MATLAB --summarize --human-readable
```

- 2 From your local installation of MATLAB, build an image containing MATLAB. This can take several minutes.

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

- 3 Check which images were loaded from the /data folder and then built by listing all MATLAB Online Server images.

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

```
ubuntu@ubuntu:~/matlab_online_server$ sudo ./mosadm list-docker-images
Listing docker images from the source docker registry: mathworks/mos with tag: R2022b
mathworks/mos/com.mathworks.matlabonlineserver.matlab-image:R2022b
...
```

Install NGINX Ingress Controller

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

Note If you have an NGINX ingress controller already in your cluster, you must update your controller settings. Because host networking is enabled, for the MATLAB Online Server configuration to work, the number of NGINX ingress controllers cannot exceed the number of nodes in the cluster.

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

```
sudo ./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
<i>your-namespace</i> -ingress-nginx-controller-7488b9897b-58czg	1/1	Running	0	54s

Prepare Helm Charts and Configure Overrides

With the cluster and Docker images ready to use, before the deploying the services, load the Helm Chart and configuration overrides. Helm is an open source project that helps deploy services into Kubernetes. For more details, see [What Is A Helm Chart? — A Beginner's Guide](#)).

- 1 Navigate to your MATLAB Online Server root folder. For example: `cd ~/matlab_online_server`.
- 2 Copy the necessary Helm charts. This command creates a folder named `charts` in the current folder and then copies the charts there.

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

- 3 Create the chart configuration files for the deployments. This command creates the YAML override files that you can use to configure the server.

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

The generated folder structure has this format:

- `./overrides/cluster/namespace/`
 - `core-ui.yaml`
 - `license.yaml`
 - `matlab-pool.yaml`
 - ...

cluster is the name of your cluster (`matlab-online-server` for single-node installations).

namespace is the value of the `NAMESPACE` parameter in your `install.config` file (default = `mathworks`).

The YAML files enable you to override the default configuration and customize the server. For example, you can configure storage and authentication. To customize the server before deployment, see “Post-Installation Additional Configuration” on page 3-2. Otherwise, proceed to the next step.

Deploy MATLAB Online Server

- 1 To install all the MATLAB Online Server components, run this command:

```
sudo ./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:


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

The pods running MATLAB can take several minutes to fully start (2/2 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” on page 5-14. Sample output:

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

Output:

NAME	READY	STATUS	RESTARTS	AGE
<i>your-namespace-core-ui-core-ui-5bd9b7c88b-fqbl4</i>	1/1	Running	0	111s
<i>your-namespace-authnz-authnz-7c7999589c-x74d9</i>	1/1	Running	0	111s
<i>your-namespace-gateway-669dd7865f-t6l74</i>	1/1	Running	0	110s
<i>your-namespace-license-548f7b6655-8skdt</i>	1/1	Running	0	108s
<i>your-namespace-matlab-pool-8c679f76-c4h4n</i>	2/2	Running	0	107s
<i>your-namespace-matlab-pool-8c679f76-cckkp</i>	2/2	Running	0	107s
<i>your-namespace-matlab-pool-ui-579fc8ff49-2bz25</i>	1/1	Running	0	107s
<i>your-namespace-ingress-nginx-controller-kcwwd</i>	1/1	Running	0	104s
<i>your-namespace-resource-resource-894447df7-r9ngb</i>	1/1	Running	0	103s

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. Run this command, replacing *domain* with the value of the `DOMAIN_BASE` parameter.

```
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 then 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 you specified in the `ML_PASSWORD` parameter of `install.config` (the default is `password`).

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](#).

Next Steps

This completes online installation.

The default configuration parameters allow a single-machine installation. These configuration settings are stored in a collection of YAML and SAML files. If you want to apply additional

configuration parameters to customize the system for one or multiple machines, see “Post-Installation Additional Configuration” on page 3-2.

Offline Install on Single Machine

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 on page 1-6.
- 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 “Resources and Links” on page 9-2 for an introduction to the terms and processes used when installing MATLAB Online Server.

Verify Installation Prerequisites

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

- Make sure you can run commands using sudo.
- The environment is running one of the Supported Platforms on page 1-6.
- Docker is installed. For the minimum version supported by MATLAB Online Server, see “Requirements” on page 1-6.
- 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.
- Verify that your operating system is hosted at a static IP address. Kubernetes requires a static IP address to function properly. With dynamic IP addresses, when the IP address changes, the Kubernetes server fails and must be reinstalled.

Install Network License Manager

Follow the Linux instructions for installing on an offline machine described in “Install License Manager on License Server” in the Help Center. 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.
- 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 will run without any errors and then will check out the license when it starts. If the license

is invalid or the license server cannot be reached, then the license service will exit and then Kubernetes will automatically attempt 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 will then reject requests with a failed status. The logs of the services indicate when they enter or exit this hibernation or grace period state.

Download MATLAB for Offline Installation

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

Follow the Linux instructions for “Install Products Using File Installation Key” in the Help Center. Make sure you install MATLAB into the folder `/MATLAB`.

Download MATLAB Online Server Installer

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

- 1 Go to the [Downloads](#) page on mathworks.com.
- 2 Under **Select Release**, select the current release, for example, R2022b.
- 3 Expand the **Get MATLAB Online Server** section. This section appears only if your account has a license for MATLAB Online Server software.
- 4 Click **Download** to download the ZIP file.
- 5 In the download folder or folder of your choice, extract the installation files. For example, this command downloads the ZIP file to your home directory (`~/`).

```
unzip R2022b_matlab_online_server.zip -d ~/
```

The unzipped contents are as follows:

- `matlab_online_server/`
 - `data/`
 - `attributions.txt`
 - `install.config`
 - `thirdpartylicenses.txt`
 - `mosadm`

The `mosadm` command is the main utility to perform the remaining installation steps. For details on the various operations it can perform, run this command:

```
./mosadm help
```

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” on page 4-9 for more details on these artifacts.

After the build command completes, check that the following folders are present:

- *matlab_online_server/data/kubepacker-direct-install-rhel-1.2.0/files/opt/mw/offlineinstaller/*
- *matlab_online_server/data/com.3p.**

Output from the build can be viewed in the terminal. If the build was not successful, try rebuilding, and if it fails again, contact [MathWorks Support](#).

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/R2022b username@server.com/MATLAB
```

- 2 Log off of the first machine and log in to the air-gapped machine.
- 3 In the *matlab_online_server* folder, open the *install.config* file and set the `OFFLINE_MODE` parameter to `true`.
- 4 Install MATLAB Online Server.

Follow the MATLAB Online Server install workflow starting with “Install Kubernetes” on page 2-3.

Next Steps

This completes the offline installation.

The default configuration parameters allow a single-machine installation. These configuration settings are stored in a collection of YAML and SAML files. If you want to apply additional configuration parameters to customize the system for one or multiple machines, see “Post-Installation Additional Configuration” on page 3-2.

Install on Cloud-Managed Kubernetes

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

Verify Installation Prerequisites

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 on page 1-6.

- 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.

- Pre-requisite for GKE only: To deploy NGINX Ingress Controller on GKE cluster, create a clusterrolebinding for the specific user as shown:

```
kubectl create clusterrolebinding cluster-admin-binding \  
> --clusterrole cluster-admin \  
> --user <userid>
```

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 then 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 on page 1-6.
- Install Helm v3.
- Install Kubectl.
- Install Docker. For the minimum version supported by MATLAB Online Server, see "Requirements" on page 1-6.

Install Network License Manager

Follow the Linux instructions for installing on an offline machine described in "Install License Manager on License Server" in the Help Center. 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.
- 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 will run without any errors and then will check out the license when it starts. If the license is invalid or the license server cannot be reached, then the license service will exit and then Kubernetes will automatically attempt 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 will then reject requests with a failed status. The logs of the services indicate when they enter or exit this hibernation or grace period state.

Download and Install MATLAB

To install MATLAB R2020a (or later) on the server, follow the Linux instructions in “Install Products Using Internet Connection” in the Help Center.

If your machine is not connected to the Internet, follow the instructions in “Install Products Using File Installation Key”.

Make sure you install MATLAB into the folder `/MATLAB`.

Download MATLAB Online Server Installer

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

- 1 Go to the [Downloads](#) page on [mathworks.com](#).
- 2 Under **Select Release**, select the current release, for example, R2022b.
- 3 Expand the **Get MATLAB Online Server** section. This section appears only if your account has a license for MATLAB Online Server software.
- 4 Click **Download** to download the ZIP file.
- 5 In the download folder or folder of your choice, extract the installation files. For example, this command downloads the ZIP file to your home directory (`~/`).

```
unzip R2022b_matlab_online_server.zip -d ~/
```

The unzipped contents are as follows:

- `matlab_online_server/`
 - `data/`
 - `attributions.txt`
 - `install.config`
 - `thirdpartylicenses.txt`
 - `mosadm`

The `mosadm` command is the main utility to perform the remaining installation steps. For details on the various operations it can perform, run this command:

```
./mosadm help
```

Configure Server

Edit the file `install.config`, located in the folder where you unzipped the installer. Before editing this file, consider creating a backup of `install.config` so that you have a record of the default settings.

In the file, modify these parameters:

Machine Configuration

Parameter	Description	Default
DOMAIN_BASE	<p>Fully qualified domain name used to access MATLAB Online Server from a web browser. For example: <code>matlabonline.mycompany.com</code>. Specifying an IP address as the domain base is not supported.</p> <p>It is recommended that the domain name is registered to your organization and that the IP address of the server or load balancer are available on the DNS server of the organization.</p> <p>To test that your domain is working, it is possible to temporarily update the host entry of your client machine to have the IP address and the fully qualified domain name. Then, you can test accessing MATLAB Online Server instance from a browser in your client using this domain name.</p>	<code>matlab.domain.com</code>

Authentication Configuration

Parameter	Description	Default
ML_PASSWORD	<p>MATLAB Online sign-in password.</p> <p>The user name for this bootstrapping mode is <code>admin</code>.</p> <p>For security purposes, it is recommended that you replace the default administration password with a more secure one.</p> <p>Identity provider configuration for authenticating MATLAB Online users is covered in “Authentication” on page 3-5.</p>	ML_PASSWORD=password

License Configuration

Parameter	Description	Default
MOS_LICENSE_SERVER	<p>MATLAB Online Server license server. Specify the port and hostname IP address in the format <code>port@host</code>.</p> <hr/> <p>Note If you have an Enterprise License, to correctly track usage, MathWorks requires two license servers: one for MATLAB Online Server (<code>MOS_LICENSE_SERVER</code> parameter) and one for MATLAB (<code>MATLAB_LICENSE_SERVER</code> parameter).</p> <p>With the two-server configuration, you must also have multiple pools (at least 2). See “Configure Multiple MATLAB Pools” on page 3-41.</p> <hr/> <p>Example: 14650@192.166.248.2</p>	"27000@flexlm"

Parameter	Description	Default
MATLAB_LICENSE_SERVER	MATLAB license server details. Specify the port and hostname IP address in the format port@host. Example: 27000@172.0.0.1	"27000@flexlm"

Remote Docker Registry Configuration

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

Parameter	Description	Default
DOCKER_REGISTRY	Remote Docker registry to push Docker images to Example: myregistry.mycompany.com	docker-registry.com
DOCKER_REPOSITORY	Remote Docker repository Example: mos	repo-prefix
IMAGE_PULL_SECRET	User-friendly name for the Kubernetes secret object. When you deploy MATLAB Online Server, this object is created from registry credentials. This secret enables the nodes to pull images from the remote registry. Example: mymossecret	mwdockerregistry

Security Configuration

Configure your environment with Transport Layer Security (TLS). Apply the settings that are appropriate for your organization.

Parameter	Description	Default
IS_TLS_ENABLED	Flag to enable or disable <u>Transport Layer Security (TLS)</u> , specified as <code>false</code> or <code>true</code> .	false
TLS_KEY_FILE	Path to the TLS key file. This parameter applies only if IS_TLS_ENABLED is set to <code>true</code> .	/opt/tls.key

Parameter	Description	Default
TLS_CERT_FILE	Path to the TLS certification file. This parameter applies only if IS_TLS_ENABLED is set to true.	/opt/tls.crt

Additional Configuration Options

You can optionally modify additional parameters based on your configuration. For a complete list of parameters, see “Installation Configuration Options for MATLAB Online Server” on page 2-32.

Prepare and Push Docker Images

- 1 Load all Docker images and files shipped through the installer onto the computer.
 - a Load Docker images:


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

Note: If your end user has access to Docker, then sudo is not required.
 - b 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
```
 - c To check which images were loaded and then built, list all MATLAB Online Server images with the following command:


```
sudo ./mosadm list-docker-images
```
- 2 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.
 - a 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 the `install.config` file.
 - b (If you are not using the Amazon ECR registry, skip this step.) Create an ECR repository. Amazon ECR registry expects repositories for all images to be created in advance. For example, you can create an ECR repository by using this command:


```
./mosadm list-docker-images | sed -E 's|mathworks/mos(.*?):.*|\1|' |
sed -n '1!p' | xargs -L 1 aws ecr create-repository --repository-name
```
 - c Push Docker images to the remote registry (configured in `install.config`) with the following command:


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

Install NGINX Ingress Controller

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

Note You must update your NGINX ingress controller settings if one already exists in your cluster. The number of NGINX ingress controllers cannot exceed the number of nodes in the cluster for the MATLAB Online Server config (hostNetwork:true) to work.

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
<i>your-namespace</i> -ingress-nginx-controller-65cd5f78f5-j8ggd	1/1	Running	0	17m

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.

Deploy MATLAB Online Server

- 1 To install all the MATLAB Online Server components, run this command:

```
sudo ./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:

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

The pods running MATLAB can take several minutes to fully start (2/2 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” on page 5-14. Sample output:

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

Output:

NAME	READY	STATUS	RESTARTS	AGE
<i>your-namespace-core-ui-core-ui-5bd9b7c88b-fqbl4</i>	1/1	Running	0	111s
<i>your-namespace-authnz-authnz-7c7999589c-x74d9</i>	1/1	Running	0	111s
<i>your-namespace-gateway-669dd7865f-t6l74</i>	1/1	Running	0	110s
<i>your-namespace-license-548f7b6655-8skdt</i>	1/1	Running	0	108s
<i>your-namespace-matlab-pool-8c679f76-c4h4n</i>	2/2	Running	0	107s
<i>your-namespace-matlab-pool-8c679f76-cckkp</i>	2/2	Running	0	107s
<i>your-namespace-matlab-pool-ui-579fc8ff49-2bz25</i>	1/1	Running	0	107s
<i>your-namespace-ingress-nginx-controller-kcwwd</i>	1/1	Running	0	104s
<i>your-namespace-resource-resource-894447df7-r9ngb</i>	1/1	Running	0	103s

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.

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. Run this command, replacing *domain* with the value of the `DOMAIN_BASE` parameter.

```
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 then 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 you specified in the `ML_PASSWORD` parameter of `install.config` (the default is `password`).

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](#).

Next Steps

This completes installation on cloud-managed Kubernetes.

The default configuration parameters allow a single-machine installation. These configuration settings are stored in a collection of YAML and SAML files. If you want to apply additional configuration parameters to customize the system for one or multiple machines, see “Post-Installation Additional Configuration” on page 3-2.

Install on Red Hat OpenShift

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

Verify Installation Prerequisites

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
```

Also complete the following software checks on the client machine before downloading and then 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 on page 1-6.
- Install Helm 3.
- Install Kubectl.
- Install Docker. For the minimum version supported by MATLAB Online Server, see "Requirements" on page 1-6.

Install Network License Manager

Follow the Linux instructions for installing on an offline machine described in "Install License Manager on License Server" in the Help Center. 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.
- 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 will run without any errors and then will check out the license when it starts. If the license is invalid or the license server cannot be reached, then the license service will exit and then Kubernetes will automatically attempt 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 will then reject requests with a failed status. The logs of the services indicate when they enter or exit this hibernation or grace period state.

Download and Install MATLAB

To install MATLAB R2020a (or later) on the server, follow the Linux instructions in “Install Products Using Internet Connection” in the Help Center.

If your machine is not connected to the Internet, follow the instructions in “Install Products Using File Installation Key”.

Make sure you install MATLAB into the folder `/MATLAB`.

Download MATLAB Online Server Installer

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

- 1 Go to the [Downloads](#) page on [mathworks.com](#).
- 2 Under **Select Release**, select the current release, for example, R2022b.
- 3 Expand the **Get MATLAB Online Server** section. This section appears only if your account has a license for MATLAB Online Server software.
- 4 Click **Download** to download the ZIP file.
- 5 In the download folder or folder of your choice, extract the installation files. For example, this command downloads the ZIP file to your home directory (`~/`).

```
unzip R2022b_matlab_online_server.zip -d ~/
```

The unzipped contents are as follows:

- `matlab_online_server/`
 - `data/`
 - `attributions.txt`
 - `install.config`
 - `thirdpartylicenses.txt`
 - `mosadm`

The `mosadm` command is the main utility to perform the remaining installation steps. For details on the various operations it can perform, run this command:

```
./mosadm help
```

Configure Server

Edit the file `install.config`, located in the folder where you unzipped the installer. Before editing this file, consider creating a backup of `install.config` so that you have a record of the default settings.

In the file, modify these parameters:

Machine Configuration

Parameter	Description	Default
DOMAIN_BASE	<p>Fully qualified domain name used to access MATLAB Online Server from a web browser. For example: <code>matlabonline.mycompany.com</code>. Specifying an IP address as the domain base is not supported.</p> <p>It is recommended that the domain name is registered to your organization and that the IP address of the server or load balancer are available on the DNS server of the organization.</p> <p>To test that your domain is working, it is possible to temporarily update the host entry of your client machine to have the IP address and the fully qualified domain name. Then, you can test accessing MATLAB Online Server instance from a browser in your client using this domain name.</p>	<code>matlab.domain.com</code>

Authentication Configuration

Parameter	Description	Default
ML_PASSWORD	<p>MATLAB Online sign-in password.</p> <p>The user name for this bootstrapping mode is <code>admin</code>.</p> <p>For security purposes, it is recommended that you replace the default administration password with a more secure one.</p> <p>Identity provider configuration for authenticating MATLAB Online users is covered in "Authentication" on page 3-5.</p>	<code>ML_PASSWORD=password</code>

License Server Configuration

Parameter	Description	Default
MOS_LICENSE_SERVER	<p>MATLAB Online Server license server. Specify the port and hostname IP address in the format port@host.</p> <hr/> <p>Note If you have an Enterprise License, to correctly track usage, MathWorks requires two license servers: one for MATLAB Online Server (MOS_LICENSE_SERVER parameter) and one for MATLAB (MATLAB_LICENSE_SERVER parameter).</p> <p>With the two-server configuration, you must also have multiple pools (at least 2). See “Configure Multiple MATLAB Pools” on page 3-41.</p> <hr/> <p>Example: 14650@192.166.248.2</p>	"27000@flexlm"
MATLAB_LICENSE_SERVER	<p>MATLAB license server details. Specify the port and hostname IP address in the format port@host.</p> <hr/> <p>Example: 27000@172.0.0.1</p>	"27000@flexlm"

Remote Docker Registry Configuration

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

Parameter	Description	Default
DOCKER_REGISTRY	<p>Remote Docker registry to push Docker images to</p> <hr/> <p>Example: myregistry.mycompany.com</p>	docker-registry.com

Parameter	Description	Default
DOCKER_REPOSITORY	Remote Docker repository Example: mos	repo-prefix
IMAGE_PULL_SECRET	User-friendly name for the Kubernetes secret object. When you deploy MATLAB Online Server, this object is created from registry credentials. This secret enables the nodes to pull images from the remote registry. Example: mymossecret	mwdockerregistry

Security Configuration

Configure your environment with Transport Layer Security (TLS). Apply the settings that are appropriate for your organization.

Parameter	Description	Default
IS_TLS_ENABLED	Flag to enable or disable Transport Layer Security (TLS) , specified as <code>false</code> or <code>true</code> .	false
TLS_KEY_FILE	Path to the TLS key file. This parameter applies only if <code>IS_TLS_ENABLED</code> is set to <code>true</code> .	/opt/tls.key
TLS_CERT_FILE	Path to the TLS certification file. This parameter applies only if <code>IS_TLS_ENABLED</code> is set to <code>true</code> .	/opt/tls.crt

Additional Configuration Options

You can optionally modify additional parameters based on your configuration. For a complete list of parameters, see “Installation Configuration Options for MATLAB Online Server” on page 2-32.

Prepare and Push Docker Images

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

- a Load Docker images:

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

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

- b 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
```

- c** To check which images were loaded and then built, list all MATLAB Online Server images with the following command:

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

- 2** 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.

- a** 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 the `install.config` file.

- b** Push Docker images to the remote registry (configured in `install.config`) with the following command:

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

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 the OpenShift ingress controller, set these fields for the version of OpenShift you are using.

OpenShift 4.7 or Earlier

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

OpenShift 4.8 or Later

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

MATLAB Online Server expects the timeout to be set to "365s" because 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 Red Hat.

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

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

Deploy MATLAB Online Server

- 1 To install all the MATLAB Online Server components, run this command:

```
sudo ./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:

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

The pods running MATLAB can take several minutes to fully start (2/2 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" on page 5-14. Sample output:

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

Output:

NAME	READY	STATUS	RESTARTS	AGE
<i>your-namespace-core-ui-core-ui-5bd9b7c88b-fqbl4</i>	1/1	Running	0	111s
<i>your-namespace-authnz-authnz-7c7999589c-x74d9</i>	1/1	Running	0	111s
<i>your-namespace-gateway-669dd7865f-t6l74</i>	1/1	Running	0	110s
<i>your-namespace-license-548f7b6655-8skdt</i>	1/1	Running	0	108s
<i>your-namespace-matlab-pool-8c679f76-c4h4n</i>	2/2	Running	0	107s
<i>your-namespace-matlab-pool-8c679f76-cckkp</i>	2/2	Running	0	107s
<i>your-namespace-matlab-pool-ui-579fc8ff49-2bz25</i>	1/1	Running	0	107s
<i>your-namespace-ingress-nginx-controller-kcwwd</i>	1/1	Running	0	104s
<i>your-namespace-resource-resource-894447df7-r9ngb</i>	1/1	Running	0	103s

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.

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. Run this command, replacing *domain* with the value of the `DOMAIN_BASE` parameter.

```
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 then 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 you specified in the `ML_PASSWORD` parameter of `install.config` (the default is `password`).

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](#).

Next Steps

This completes installation on Red Hat OpenShift.

The default configuration parameters allow a single-machine installation. These configuration settings are stored in a collection of YAML and SAML files. If you want to apply additional configuration parameters to customize the system for one or multiple machines, see “Post-Installation Additional Configuration” on page 3-2.

Installation Configuration Options for MATLAB Online Server

MATLAB Online Server includes a configuration file that you can use to configure your installation. This file is named `install.config` and is located in the root of your MATLAB Online Server installation. For example, if you installed the server in your home directory, then it located at `~/matlab_online_server/install.config`.

The tables that follow lists the parameters you can modify or enable. If you edit this file while your server is running, you must redeploy the server for your changes to take effect.

Domain Name

Parameter	Description	Default
DOMAIN_BASE	<p>Fully qualified domain name used to access MATLAB Online Server from a web browser. For example: <code>matlabonline.mycompany.com</code>. Specifying an IP address as the domain base is not supported.</p> <p>It is recommended that the domain name is registered to your organization and that the IP address of the server or load balancer are available on the DNS server of the organization.</p> <p>To test that your domain is working, it is possible to temporarily update the host entry of your client machine to have the IP address and the fully qualified domain name. Then, you can test accessing MATLAB Online Server instance from a browser in your client using this domain name.</p>	<code>matlab.domain.com</code>

Docker

If you have a remote registry within your organization, uncomment these parameters and configure the following information about the registry.

Parameter	Description	Default
DOCKER_REGISTRY	Remote Docker registry to push Docker images to Example: myregistry.mycompany.com	docker-registry.com
DOCKER_REPOSITORY	Remote Docker repository Example: mos	repo-prefix
IMAGE_PULL_SECRET	User-friendly name for the Kubernetes secret object. When you deploy MATLAB Online Server, this object is created from registry credentials. This secret enables the nodes to pull images from the remote registry. Example: mymossecret	mwdockerregistry

This Docker parameter is enabled by default.

Parameter	Description	Default
IMAGE_TAG	Release tag of Docker image.	<i>R/NNNNx</i> , which is the MATLAB Online Server release, such as

Namespace

Parameter	Description	Default
NAMESPACE	Namespace to use for the deployment.	mathworks

Authentication

Parameter	Description	Default
ML_PASSWORD	<p>MATLAB Online sign-in password.</p> <p>The user name for this bootstrapping mode is <code>admin</code>.</p> <p>For security purposes, it is recommended that you replace the default administration password with a more secure one.</p> <p>Identity provider configuration for authenticating MATLAB Online users is covered in "Authentication" on page 3-5.</p>	ML_PASSWORD=password

License Server

Parameter	Description	Default
MOS_LICENSE_SERVER	<p>MATLAB Online Server license server. Specify the port and hostname IP address in the format port@host.</p> <hr/> <p>Note If you have an Enterprise License, to correctly track usage, MathWorks requires two license servers: one for MATLAB Online Server (MOS_LICENSE_SERVER parameter) and one for MATLAB (MATLAB_LICENSE_SERVER parameter).</p> <p>With the two-server configuration, you must also have multiple pools (at least 2). See “Configure Multiple MATLAB Pools” on page 3-41.</p> <hr/> <p>Example: 14650@192.166.248.2</p>	"27000@flexlm"
MATLAB_LICENSE_SERVER	<p>MATLAB license server details. Specify the port and hostname IP address in the format port@host.</p> <hr/> <p>Example: 27000@172.0.0.1</p>	"27000@flexlm"

Security

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.

Parameter	Description	Default
IS_TLS_ENABLED	Flag to enable or disable Transport Layer Security (TLS) , specified as false or true.	false

Parameter	Description	Default
TLS_KEY_FILE	Path to the TLS key file. This parameter applies only if IS_TLS_ENABLED is set to true.	/opt/tls.key
TLS_CERT_FILE	Path to the TLS certification file. This parameter applies only if IS_TLS_ENABLED is set to true.	/opt/tls.crt

Kubernetes

Parameter	Description	Default
KUBERNETES_CERTIFICATE_NAME	Name of a pre-existing Kubernetes certificate	"" (empty string)

Base Container Image

Parameter	Description	Default
BASE_IMAGE_FLAVOR	The base container images that are used to install and configure MATLAB Online Server, specified as <code>debian</code> or <code>rhel</code> .	debian

Offline Installation

Parameter	Description	Default
OFFLINE_MODE	Run MATLAB Online Server in offline mode, specified as <code>true</code> or <code>false</code> . For more details, see "Offline Install on Single Machine" on page 2-11.	false

Client-Only Mode

Parameter	Description	Default
CLIENT_ONLY	Run MATLAB Online Server in client-only mode, specified as <code>false</code> or <code>true</code> .	false

Logging

Parameter	Description	Default
LOG_LEVEL	Log level of MATLAB Online Server, specified as <code>info</code> or <code>debug</code> . For more verbose logs, specify <code>LOG_LEVEL=debug</code> . For more verb	<code>info</code>

High Availability

Parameter	Description	Default
IS_HA_ENABLED	Option to enable or disable high availability (HA) for MATLAB Online Server, specified as <code>false</code> or <code>true</code> . For more details, see "MATLAB Online Server in an HA Environment" on page 7-21.	<code>false</code>

Dashboard Deployment

Parameter	Description	Default
DEPLOY_DASHBOARDS	Option to deploy dashboards, specified as <code>false</code> or <code>true</code> .	<code>false</code>

Firewall

Parameter	Description	Default
FIREWALL_ENABLED	<p>Option to enable or disable a firewall, specified as true or false.</p> <p>This parameter applies only to single-node installations. For more details, see:</p> <ul style="list-style-type: none"> “Online Install on Single Machine” on page 2-2 “Offline Install on Single Machine” on page 2-11 <p>When this firewall is enabled, only ports 22 (SSH), 80 (HTTP) and 443 (HTTPS) are enabled. To communicate on any other port, you must use the ufw (uncomplicated firewall command). For example, this command enables communication on port 8443:</p> <pre>sudo ufw allow 8443</pre>	true

See Also

Related Examples

- “Online Install on Single Machine” on page 2-2
- “Offline Install on Single Machine” on page 2-11
- “Install on Cloud-Managed Kubernetes” on page 2-14
- “Install on Red Hat OpenShift” on page 2-23

Post-Installation Additional Configuration

Post-Installation Additional Configuration

Follow the directions in these topics to customize the configuration for your implementation of MATLAB Online Server.

Options include:

- “Customize Organization Name and Logo” on page 3-3
- “Authentication” on page 3-5 Settings
- “Configure MATLAB Pool” on page 3-14 and “Configure Multiple MATLAB Pools” on page 3-41
- “Hardware Resources” on page 3-18
- “Specify MATLAB Documentation” on page 3-43

See Also

Related Examples

- “Introduction to MATLAB Online Server” on page 1-2

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 “Authentication” on page 3-5. The default sign-in page for MATLAB Online, with no customization, is shown in the following image:

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

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"
```

Specify Logo

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

Link to Logo

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

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

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
```

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 Server on MATLAB Online Server:



Authentication

To authenticate user access in MATLAB® Online Server™, you must configure an identity provider that MATLAB Online Server can use to authenticate and allow users. The identity provider allows the use of user credentials for the organization through a user name/password prompt or single sign-on, depending on the type of Identity provider configured.

At least one identity provider must be configured to sign in to MATLAB Online Server. When you issue the command to generate overrides, the command generates a configuration for the identity provider of type local (see “Local” on page 3-6 for details) with the user name `admin` and password that are defined in `install.config`.

To apply the identity provider configuration, edit the settings in the file `<matlab_online_server_install_dir>/overrides/<cluster>/<namespace>/authnz.yaml`. For example, `/opt/matlab_online_server/overrides/matlab-online-server/mathworks/authnz.yaml`.

Identity Provider

Identity Provider Specification

Attribute	Default Value	Optional	Description
<code>identityProviders</code>	<code>[]</code>	No	The list of identity providers configured for use with MATLAB Online Server

In YAML format:

```
identityProviders:
- <list of identity provider configurations>
```

Attribute	Default Value	Optional	Description
<code>name</code>	<code>""</code>	No	The name of the identity provider
<code>displayName</code>	<code>""</code>	Yes	How the identity provider's name is displayed, specified using characters A-Z, a-z, and 0-9, with the symbols <code>_</code> , <code>-</code> , or <code>.</code> in between these characters. The maximum <code>displayName</code> can be 63 characters long.
<code>type</code>	<code>""</code>	No	The type of the identity provider. Based on the type, the identity provider properties vary; see next section.

Identity Provider Types

MATLAB Online Server supports the following types of identity providers:

- LOCAL

- LDAP
- SAML

Local

The LOCAL (**local**) identity provider type sets up the accounts that are in memory. For this identity provider, the accounts must be defined with the properties shown in the following table.

Attribute	Default Value	Optional	Description
subjectId	""	No	The userID or user name for the account
displayName	""	No	Display name for the account
password	""	No	Account password
groups	[]	Yes	The groups that the user is associated with For example: ["group1", "group2"]
extra	{}	Yes	The extra metadata that may be added to the user account when they sign in. For example: { "uid": "1001", "department": "MATLAB Online Server" }

Values in the settings for groups and extra are user-defined and not validated in any way.

In YAML format:

```
identityProviders:
  - id: local
    type: local
    displayName: local
    accounts:
      - subjectId: admin
        displayName: Admin
        password: admin
        groups: ["admin"]
        extra: {}
      - subjectId: stateful
        password: stateful
        extra: {}
```

LDAP

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. The LDAP (**ldap**) identity provider type 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.

For more about LDAP settings, visit <https://ldap.com/>.

The LDAP identity provider type requires extra properties, shown in the following table:

Attribute	Default Value	Optional	Description
host	""	No	The host address for the LDAP, for example: "ldapservers.yourcompany.com".
port	""	No	Port is the LDAP port at the host address, for example: "389" for regular LDAP and "636" for secure LDAP.
useStartTls	false	Yes	The useStartTLS attribute is a Boolean flag that indicates if your LDAP server requires start TLS for connection. You must add the trusted certificates to the MATLAB Online Server trust store. See "Add Certificates to MATLAB Online Server Trust Store" on page 3-10.
useSsl	false	Yes	The useSSL attribute is a Boolean flag that indicates if your LDAP server requires SSL for connection. You must add the trusted certificates to the MATLAB Online Server trust store. See "Add Certificates to MATLAB Online Server Trust Store" on page 3-10.
baseDn	""	No	The baseDn attribute is the starting point for the search tree in your LDAP environment. For example: "dc=ldap,dc=yourcompany,dc=com"

Attribute	Default Value	Optional	Description
filter	""	No	<p>Specifies an LDAP search filter. Use the search filter to scan a subset of users from the LDAP database. This allows you to reduce the scope of the LDAP search and control which organizational user may access MATLAB Online Server.</p> <ul style="list-style-type: none"> You can leverage variable substitution to configure how the username is provided to LDAP, for example: "(CN={0})". Specify the search filter as attribute=value, for example, CN=test*, matches all users that have a common name (CN) attribute that starts with "test". Use parentheses to combine multiple filter expression in an AND (&) or OR() clause. For example, "(&(CN={0})(department=foo))" matches all users that are in department "foo". <p>The default search filter is objectClass=organizationalPerson.</p> <p>For more information on search filters, see LDAP filters at https://ldap.com/ldap-filters.</p>
adminDn	""	No	<p>The administrator account used to establish the connection to the LDAP server, for example: "cn=readonly".</p> <p>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.</p>
adminPassword	""	No	<p>The administrator account password used to establish the connection to the LDAP server.</p>

Attribute	Default Value	Optional	Description
subjectAttributeMapping	{}	No	<p>The subject attribute mapping allows configurations to set the values for the user attributes that map to your internal LDAP values.</p> <p>For example, your user's first name might be stored as <i>displayFirstName</i> in the LDAP system.</p> <p>To map this in MATLAB Online Server set <code>displayName: displayFirstName</code>.</p> <p>Subject Map</p> <pre>subjectAttributeMapping: subjectId: displayName: extra: uid:</pre> <p><code>displayName</code> is an optional attribute that controls what is displayed for the user name in the upper right-hand corner of MATLAB Online Server.</p> <p>The <code>subjectId</code> is the default mapping field for the user account, normally "cn".</p> <p>The extra parameters are for additional customization relating to the network files system (NFS):</p> <ul style="list-style-type: none"> • <code>extra.uid</code> - the mapping to a Unix UID id

In YAML format:

```
identityProviders:
- id: "ldap"
  type: "ldap"
  displayName: "LDAP"
  host: "ldap"
  port: "636"
  useStartTls: true
  useSsl: true
  baseDn: "dc=matlabonlineserver,dc=mwcloudtest,dc=com"
  filter: "(uid={0})"
  adminDn: "cn=readonly,dc=matlabonlineserver,dc=mwcloudtest,dc=com"
  adminPassword: "readonly"
  subjectAttributeMapping:
    subjectId: "uid"
    displayName: "cn"
    groups: "groups"
```

```
extra:
  uid: "uidNumber"
```

Add Certificates to MATLAB Online Server Trust Store

When you use LDAPS or LDAP with TLS, you must add the certificates to the global trust store. These certificates are validated against the LDAP/LDAPS server to establish a secure connection.

You must add the certificate information, shown in the following table, to authnz.yaml.

Attribute	Default Value	Optional	Description
validateCertificate	true	Yes	<p>validateCertificate property enables the certificate verification for the authnz service while communicating to the external identity providers</p> <p>As an example, when LDAP is being used with useSSL= true or useTLS = true, the secure communication will be validated.</p> <p>If this property is disabled, the communication will still be LDAPS, but the certificate verification will be skipped.</p>
trustedCertificates	""	Yes	<p>The server certificates that are required for communicating with the external identity providers in the authnz service must be provided using the trustedCertificates property.</p> <p>For example, when LDAP is being used with useSSL= true or useTLS = true, the secure communication must be validated and server certificates must be provided using this property.</p> <p>If the certificates are not provided, the LDAPS communication fails.</p>

In YAML format:

```
global:
  tls:
    validateCertificate: true
    trustedCertificates: |
      -----BEGIN CERTIFICATE-----
      MIIC5DCCAk2gAwIBAgIUZhnmE MmayNSTcAI2hgyxQt6GotUwDQYJKoZIhvcNAQEL
      ...
      VA/d/fQ+yxUjLDBc6ly/0wVFtIr0QyKE
      -----END CERTIFICATE-----
      -----BEGIN CERTIFICATE-----
      BQAwgYMxCzAJBgNVBAYTALVTMQswCQYDVQQIDAJNQTEPMA0GA1UEBwwGTmF0aWNr
```



```

...
EdUg4pCYWUyFgGA/QCg4EniQEMN
-----END CERTIFICATE-----

```

The certificates must be formatted using PEM (see “PEM” on page 9-2).

SAML

Security Assertion Markup Language, also known as SAML, is used to communicate with user identities. The SAML configuration for MATLAB Online Server ensures that the identities used to control access to MATLAB Online Server are managed by the Enterprise Identify Provider.

For more information about SAML settings, see <https://wiki.oasis-open.org/security/>.

The SAML identity provider type requires extra properties, as shown in the following table:

Attribute	Default Value	Optional	Description
assertionConsumerPath	""	No	The assertion consumer service (ACS) endpoint is a location where the SSO tokens are sent. ACS is applicable to all SAML versions and both the IdP- and SP-initiated SSO profiles.
corsAllowOriginDomain	""	No	The full domain name of the server that communicates with MATLAB Online Server, for example, https://samlintegrationhost.yourcompanydomain.com
relyingPartyId	""	No	According to the SAML specification, the value of relyingPartyId specifies the host name for the server that integrates with the SAML Identity Provider. As a result, this value is 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 expose the Identity Provider via the SAML, this may be an arbitrary identifier.

Attribute	Default Value	Optional	Description
idpMetadataXml	""	No	<p>The value of idpMetadataXml is a block of xml that is exported by the SAML Identity Provider. This block of xml can just be pasted in to the authnz.yaml file. However, note that this block must be properly formatted YAML, otherwise there will be problems deploying AuthNZ via the mosadm utility.</p> <p>Depending on the integration product or service being used to expose the Identity Provider via SAML, there might be newline characters. Delete those newline characters from the metadata xml before pasting the code into authnz.yaml.</p>
subjectAttributeMapping	{}	No	<p>The subject attribute mapping lets you set values for the user attributes that map to your organizational SAML assertion values.</p> <p>For example, your user's first name might be stored as <i>displayFirstName</i> in the SAML system.</p> <p>To map this in MATLAB Online Server, set <code>displayName: displayFirstName</code>.</p> <p>Subject Map</p> <pre style="border: 1px solid black; padding: 5px;">subjectAttributeMapping: subjectId: displayName: extra: uid:</pre> <p><code>displayName</code> is an optional attribute that controls what is displayed for the user name in the upper right-hand corner of MATLAB Online Server.</p> <p><code>subjectId</code> is the default mapping field for the user account, normally "cn".</p> <p>The extra parameters are for additional customization relating to the network files system (NFS):</p> <ul style="list-style-type: none"> • <code>extra.uid</code> - the mapping to a Unix UID id

In YAML format:

```
identityProviders:
- type: saml
  assertionConsumerPath: "/service/assertionConsumer"
  corsAllowOriginDomain: "https://samlintegrationhost.yourcompanydomain.com"
```

```
relyingPartyId: "https://matlab.domain.com"
idpMetadataXml: |
  <xml content/>
subjectAttributeMapping :
  displayName: "name"
  groups: "groups"
  extra :
  uid:
```

Configure MATLAB Pool

The MATLAB Pool settings file (`matlab-pool.yaml`) determines how MATLAB users, or groups of users, interact with the online server. It is located at this path:

```
overrides/cluster/namespace/matlab-pool.yaml
```

where:

- `cluster` is the name of your Kubernetes cluster.
- `namespace` is the Kubernetes namespace you used to deploy MATLAB Online Server.

For example:

```
overrides/matlab-online-server/mathworks/matlab-pool.yaml
```

By modifying the YAML fields in this file, you can override the default settings, such as the number of MATLAB pods ready to use at login and the installation source of the MATLAB pool.

When you update any `matlab-pool.yaml` settings, you must update the MATLAB Pool by issuing this command:

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

License Server

To configure the server used to check out MATLAB licenses, update these fields in the `matlab-pool.yaml` file:

```
flexlm:  
  servers:
```

The MathWorks License Manager uses [FlexNet Publisher](#) (formerly FLEXlm) to check for licenses.

In the `servers` subfield of `flexlm`, list the license server in the format `port@address`. For example:

```
flexlm:  
  servers: "27000@flexlm.company.com"
```

To list multiple servers, separate the server information with a colon. For example:

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

MATLAB Pods Ready at Login

To configure how many MATLAB Pools to keep available and ready to assign to users when they log in, update the `replicaCount` field of `matlab-pool.yaml`.

```
replicaCount: 2
```

The default count of 2 means that two MATLAB Pool pods are ready for assignment. For example, suppose two users log in at the same time. Each pod assigned to them contains a MATLAB instance

that is ready to use within seconds of logging in. If two more users log in at the same time, their logins are delayed until two new pods can start up.

The maximum number of active users that can log in at one time is limited to the number of MATLAB licenses. The total number of MATLAB instances that can be assigned is limited to the number of licenses available and the hardware configured for the MATLAB Pool. If no MATLAB instances are available, users receive an error at login.

MATLAB Installation Source

Specify the source of the MATLAB Pool pod installation in the `mountType` field of the `matlab-pool.yaml` file. MATLAB Online Server uses this information to mount the MATLAB installation so that it can interact with it.

This table shows the available configuration options.

MATLAB Installation Source	mountType Value	Description
Container image	"embedded" (default)	<ul style="list-style-type: none"> Enables easy export of MATLAB installation across Kubernetes clusters. Requires large image installation. <p>See "Mount MATLAB from Built Container Image" on page 3-16.</p>
One or more Kubernetes nodes	"host"	<ul style="list-style-type: none"> Enables use of existing MATLAB installation, without a large image installation. Export across Kubernetes clusters is more difficult. <p>See "Mount MATLAB from Kubernetes Nodes" on page 3-16.</p>
NFS server	"nfs"	<ul style="list-style-type: none"> Enables use of existing MATLAB installation, without a large image installation. Performance might be slower because installation is on a network drive. <p>See "Mount MATLAB from NFS Server" on page 3-17.</p>

MATLAB Installation Source	mountType Value	Description
Persistent volume claim (PVC)	"pvc"	<ul style="list-style-type: none"> • Can be used with cloud-provided Kubernetes clusters. • Performance might be slower because installation is on a network drive. <p>See "Mount MATLAB from Persistent Volume Claim" on page 3-17.</p>

Mount MATLAB from Built Container Image

With this mounting configuration, you build a MATLAB container image and make the image available inside MATLAB Online Server. This is the default option.

Use this configuration to make it easy to export the MATLAB installation across Kubernetes clusters, such as when:

- Exporting to different cloud-provided Kubernetes clusters
- Working across different Kubernetes clusters within the same version of MATLAB

For details on building a MATLAB image in the container, see "Build MATLAB Image" on page 4-4.

To use this configuration, set these `matlab-pool.yaml` fields:

```
matlab:
  installation:
    mountType: "embedded"
```

To specify the container image that MATLAB Online Server uses, set these fields:

```
images:
  matlabEmbedded:
    image: "com.mathworks.matlabonlineserver.matlab-image"
    tag: "R2022b"
    registry:
    repository:
```

- Update `image` and `tag` with the information for your image.
- To use the global image settings, leave `registry` and `repository` empty.

Mount MATLAB from Kubernetes Nodes

If MATLAB is installed on the same machine as your single-node or multinode Kubernetes cluster, you can make the MATLAB installation available to MATLAB Online Server directly.

For cloud-provided Kubernetes installations, unless the nodes are customized to support this configuration, this mounting type is not recommended.

To use this configuration, set these `matlab-pool.yaml` fields:

```
matlab:
  installation:
    mountType: "host"
    path: "/MATLAB"
```

- Update `path` with the path to the MATLAB installation folder. For multinode Kubernetes installations, the `path` value must be the same across all nodes.

Mount MATLAB from NFS Server

Use this configuration when the MATLAB installation is on an NFS server that the Kubernetes cluster can access. This option makes the MATLAB installation directly available to MATLAB Online Server without having to build a large image.

Performance might be slower with this option when compared to the direct installation options, because MATLAB Online Server loads MATLAB from a network location.

To use this configuration, set these `matlab-pool.yaml` fields:

```
matlab:
  installation:
    mountType: "nfs"
    server: "nfs-server"
    path: "/exports/MATLAB/R2022b"
```

- Update `server` with the IP address of the NFS server or a fully-qualified server name.
- Update `path` with the subpath on the NFS server that contains the MATLAB installation.

Mount MATLAB from Persistent Volume Claim

If the previous configurations are not available, you can make the MATLAB installation available through a Kubernetes Persistent Volume Claim (PVC). You can back the PVC with a different storage provider by using a Container Storage Interface (CSI). For more details on PVC mounting, see "Persistent Volume Claim (PVC)" on page 3-24.

Consider using this option for cloud-provided Kubernetes installations. Performance might be slower with this option when compared to the direct installation options, because MATLAB Online Server loads MATLAB from a network location.

To use this configuration, set these `matlab-pool.yaml` fields:

```
matlab:
  installation:
    mountType: "pvc"
    claimName: "matlab-r2022b-pvc"
    path: ""
```

- Update `claimName` with the name of the PVC where the MATLAB installation is available. For multinode Kubernetes clusters, the PVC must be available to all nodes.
- Update `path` with the path within the PVC where the MATLAB installation is available.

Hardware Resources

When new MATLAB and storage use cases arise, you might need to change the hardware resources allocated to MATLAB Online Server service. To change the hardware resource allocation, update the memory and CPU configuration settings in the `matlab-pool.yaml` file.

The MATLAB pool has two containers that control the hardware resources allocated:

- The Resource-Proxy container provisions the storage for each user session.
- The MATLAB container assigns MATLAB compute resources to each user.

The default memory and CPU configuration for these containers are as follows:

```
resourceQuota:
  matlab-pool-resource-proxy:
    requests:
      cpu: "200m"
      memory: "150Mi"
    limits:
      cpu: "500m"
      memory: "512Mi"
  matlab-pool-matlab:
    requests:
      cpu: "200m"
      memory: "2Gi"
    limits:
      #cpu:
      #memory:
```

With these default settings:

- The Resource-Proxy container requests a minimum of 200m CPU and 150Mi of memory, with limits of 500m CPU and 512Mi of memory on the node.
- The MATLAB container requests a minimum of 200m CPU and 2Gi of memory on the node, with no limit set on these resources. To add resource limits, remove the comment markers (#) and add values appropriate for your environment and infrastructure. For more details, see “Add Limits to MATLAB Container” on page 3-18.

For more about CPU and memory configuration in Kubernetes, see these 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>

Add Limits to MATLAB Container

Because the default MATLAB container configuration has no resource limits specified, it is possible for a container to use up all CPU and memory on the node.

To set a limit on CPU and memory requests, in the `matlab-pool.yaml` file, uncomment the `cpu` and `memory` fields for the MATLAB container and set CPU and memory values appropriate for your environment and infrastructure. For example:


```
resourceQuota:
  matlab-pool-matlab:
    requests:
      cpu: "200m"
      memory: "2Gi"
    limits:
      cpu: "2000m"
      memory: "7Gi"
```

With these settings, the MATLAB container requests a minimum of 200m CPU and 2Gi of memory and can request a maximum of 2000m CPU and 7Gi of memory. Therefore, the container cannot use all the resources on the node.

Increase Limits of Resource-Proxy

If the amount of data that is mounted through storage profiles is excessive, such as more than 10 TB, the underlying Linux storage mechanism, FUSE, might need more memory to successfully provision the storage and required permissions.

To increase the limit on requests for CPU and memory, in the `matlab-pool.yaml` file, specify new `cpu` and `memory` field values that are appropriate for your environment and infrastructure. For example:

```
resourceQuota:
  matlab-pool-resource-proxy:
    requests:
      cpu: "250m"
      memory: "300Mi"
    limits:
      cpu: "750m"
      memory: "1Gi"
```

With these settings, the Resource-Proxy container requests a minimum of 250m CPU and 300Mi of memory and can request a maximum of 750m CPU and 1Gi of memory. Therefore, the storage provisioner can use more hardware resources than the default maximum of 500m CPU and 512Mi of memory.

Storage

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.yaml`. These 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” on page 3-29.

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. See example and more below this table.
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 then made available to the MATLAB user (see "Mount Specifications" on page 3-22).
addons	<Nil>	Yes	Directory where MATLAB add-ons are stored (see "Specification for Add-Ons" on page 3-22).

Attribute	Default Value	Optional	Note
allowedGroups	<Nil>	Yes	<p>List of the groups that user should belong to so that they can use this profile</p> <p>These groups are compared against the list of groups that are configured through <code>authnz</code> (see “Placeholders” on page 3-29).</p> <ul style="list-style-type: none"> • If <code>allowedGroups</code> is empty (default), the storage profile can be mounted by any user. • If there are multiple values in <code>allowedGroups</code>, a user that is part of any group listed can mount this storage profile. • The complete/exact group name that is returned in the <code>SubjectAuthResponse</code> should be included in <code>allowedGroups</code> as well. <p>For example, in the LDAP scenario, the group name is the full Distinguished Name (DN): (CN=group2,OU=Group,OU=Mail,DC=ad,DC=domain,DC=com)</p>

In YAML format:

```
storage:
  profiles:
    - name: <name>
      mounts:
        <list of mounts>
      startDirectory: <start directory>
      restrictNavigationOutsideMounts: <true/false>
      addons:
        directory: <addons directory>
      allowedGroups:
        - <name_of_group_1>
        - <name_of_group_2>
```

startDirectory—User session and Preferences directory. This folder contains the preferences, settings, history, and layout files associated with the user. For more about MATLAB user settings, see the MATLAB documentation in the Help Center.

MATLAB Online Server provides the ability to store the MATLAB preferences and some of the session information as part of the storage profile configuration.

For each storage profile, `startDirectory` can be customized. This field sets the session directory when the user signs into MATLAB Online Server.

When this directory is persistent, the history, layout preferences, and more will be automatically restored when the user next signs in.

Format:

```
startDirectory: "/home/${subject.subjectId}"
```

Specification for Add-Ons

Attribute	Default Value	Optional	Note
directory	<startDirectory>/ MATLAB Add-Ons	Yes	Directory where MATLAB add-ons are stored.

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" (next).
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" (next).

Attribute	Default Value	Optional	Note
createIfNotExist <ul style="list-style-type: none"> This feature is applicable only when the dynamic placeholders are used . This feature is applicable only when the permissionType is of "User." 	false	Yes	Dynamically create folders while provisioning the storage for MATLAB user, if the folder does not exist with the same UID permissions of the user.

Mount Types

MATLAB Online Server supports the following mount types:

- Empty
- Host
- NFS
- PVC

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>
```

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>
```

NFS

The NFS (**nfs**) mount type mounts an NFS directory and then 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,wsiz=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: <permission type>
          startDirectory: <start directory>
          restrictNavigationOutsideMounts: <true/false>
          addons:
            directory: <addons directory>
```

Persistent Volume Claim (PVC)

PVC (**pvc**) mount type mounts a Persistent Volume Claim that is available on the Kubernetes cluster and makes it available to the MATLAB user.

Refer to the following link for detailed information on Persistent Volumes: <https://v1-22.docs.kubernetes.io/docs/concepts/storage/persistent-volumes>

Note MATLAB Online Server expects that the Persistent Volume Claim is allowed for:

- Mounting onto the MATLAB pods (same PVC)
 - Some PV types might restrict the scaling based on the PV configuration
- Read and write files
- Retaining data after exiting the session (using `persistentVolumeReclaimPolicy`)

If Persistent Volume or Persistent Volume Claim is created in Kubernetes with a different configuration, you must reconfigure the object to satisfy the above conditions.

This mount type requires the following extra properties:

- `claimName`
 - The name of the Persistent Volume Claim
- `subPath`
 - The folder within the data volume (Persistent Volume Claim) that is available for the MATLAB user

These extra properties are shown in YAML format in the following example:

```
storage:
  profiles:
    - name: <name>
      mounts:
        - name: home
          mountPath: /share
          type: pvc
          claimName: <pvc name>
          subPath: "/exports/home/${subject.subjectId}"
          permissionType: <permission type>
          startDirectory: <start directory>
          restrictNavigationOutsideMounts: <true/false>
      addons:
        directory: <addons directory>
```

Some important points to keep in mind when using Persistent Volume Claim:

- The same Persistent Volume Claim cannot be used twice in the same Storage Profile.
- The same Persistent Volume Claim defined in the storage profile is mounted in all MATLAB pods.
- The storage resources request that is defined in the Persistent Volume or Persistent Volume Claim must be sufficient to meet the entire storage needs for all users.

Permission Types

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

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>
```

Fixed

The Fixed (**fixed**) permission type performs a mount and then 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,wsiz=1048576,namlen=255,acreg
            min=600,acregmax=600,acdirmin=600,acdirmax=600,hard,nocto,noacl,
            proto=tcp,timeo=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>

```

User

The User (**user**) permission type performs a mount and then 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” on page 3-29.

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/${subject.subjectId}"
      mountOptions:
"rw,relatime,vers=3,rsize=1048576,wsiz=1048576,namlen=255,acreg
min=600,acregmax=600,acdirmin=600,acdirmax=600,hard,nocto,noacl,
proto=tcp,timeo=600,retrans=2,mountproto=tcp,local_lock=none,nol
ock"
      permissionType: user
      uid: "${subject.uid}"
      gid: "${subject.gid}"
      startDirectory: <start directory>
      restrictNavigationOutsideMounts: <true/false>
      addons:
        directory: <addons directory>
```

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"
```

Storage Profiles: Dynamic Folder Creation

Currently, MATLAB Online Server provides a way to mount the file system available to the MATLAB user with the dynamic placeholders support of storage profiles. This feature extends the ability to create the folders automatically in their storage mount with the appropriate user permissions.

With the `createIfNotExist` attribute, you can dynamically create folders while provisioning the storage for the user, if the folder does not exist with the same UID permissions of the user.

For example, assume the following storage profile is defined and is applicable for the user when they sign in:

```

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

```

- The placeholder "\${subject.subjectId}" is resolved to the user name, for example, username1, when a user signing in to MATLAB Online Server as username1. The NFS server and the subpath "nfs:/exports/home" are mounted by the storage service
- If the folder /exports/home/username1 exists in nfsserver "nfs," then the storage service uses that folder. Otherwise, the folder is created after placeholder resolution with the permissions "\${subject.uid}" (for example, 3904).
- The folder permissions on the Linux file system (in the NFS server) are:

```

ls -al /exports/home
drwxr-xr-x 1 3904 3904 20 Feb 18 19:13 /exports
/home/username1/

```

The createIfNotExist attribute is applicable only when:

- Dynamic placeholders are used
- permissionType is "User"

```

ls -al /exports/home
drwxr-xr-x 1 3904 3904 20 Feb 18 19:13 /exports
/home/username1/

```

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 "Authentication" on page 3-5.

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": [
      "group1",
      "group2"
    ],
    "extra": {
      "uid": [
        "3904"
      ],
      "gid": [
        "3904"
      ]
    },
    "idp": "ldap"
  },
  "authToken": "xxxxxxxx-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,wsiz=1048576,namlen=255,acreg
            min=600,acregmax=600,acdirmin=600,acdirmax=600,hard,nocto,noacl,
            proto=tcp,timeo=600,retrans=2,mountproto=tcp,local_lock=none,nol
            ock"
            restrictNavigationOutsideMounts: false
```

Note that mountPath cannot be started with a placeholder.

Storage Profiles: Examples

The following examples demonstrate creating storage profiles based on the following types of storage systems (backed by Kubernetes Persistent Volumes):

- NFS
- AWS
 - EBS
 - EFS
- Azure
 - Azure Files

Results of these examples may vary based on variable information, such as Kubernetes version, features on the Kubernetes cluster, updates by the cloud provider that are more recent than these examples, and so on.

Each example includes the following:

- Pre-requisites
 - Persistent Volumes, Persistent Volume Claims needs the appropriate storage class (CSI Driver) to get created successfully
 - Refer to the following link for detailed information on Persistent Volumes: <https://v1-22.docs.kubernetes.io/docs/concepts/storage/persistent-volumes>
- Create Persistent Volume Claim
 - Provides a sample yaml file that can be used to create the Persistent Volume Claim and other dependent objects, if required.
 - Provides a command to create these objects in Kubernetes.
- Storage Profile
 - 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
```
 - After MATLAB Online Server is deployed successfully with the new storage profiles, when a MATLAB user signs in to MATLAB Online Server, the working directory of the user will be /home/<username>.
- Cleanup (optional)
 - Provides commands for removing the Persistent Volume Claim and the dependent Kubernetes objects, if needed

NFS - PVC Storage Profile

The following example provides instructions to create a Persistent Volume Claim where the data is stored in an NFS server. This newly created Persistent Volume Claim in MATLAB Online Server can provide data persistence to the MATLAB user upon signing in.

1 Pre-Requisites

The cluster must install the required NFS CSI driver. There are many Kubernetes CSI drivers that are available, and the instructions differ based on which is installed.

This example uses the default NFS CSI driver created by the Kubernetes project:

a Add helm repo:

```
helm repo add csi-driver-nfs https://raw.githubusercontent.com/kubernetes-csi/csi-driver-nfs/master/charts
```

b Install CSI driver:

```
helm install csi-driver-nfs csi-driver-nfs/csi-driver-nfs --namespace kube-system
```

2 Create Persistent Volume Claim

Create Persistent Volume Claim using YAML and kubectl.

- Save the following lines in a YAML file, for example, "nfs-pv.yaml."

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-example-nfs
spec:
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteMany
  persistentVolumeReclaimPolicy: Retain
  mountOptions:
    - relatime
    - vers=3
    - rsize=1048576
    - wsize=1048576
    - namlen=255
    - acregmin=600
    - acregmax=600
    - accdirmin=600
    - accdirmax=600
    - hard
    - nocto
    - noacl
    - proto=tcp
    - timeo=600
    - retrans=2
    - mountproto=tcp
    - local_lock=none
    - nolock
  csi:
    driver: nfs.csi.k8s.io
    readOnly: false
    volumeHandle: unique-volumeid # make sure it's a unique id in the cluster
    volumeAttributes:
      server: examplenfsserver
      share: /local-ssd/nfs_pv
---
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-example-nfs
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
  volumeName: pv-example-nfs
  storageClassName: ""
```

- Create the Persistent Volume and Persistent Volume Claim using the following command:

```
kubectl apply -n mathworks -f nfs-pv.yaml
```

3 Storage Profile

```
storage:
  profiles:
  - name: home
    mounts:
    - name: home
      mountPath: "/home/${subject.subjectId}"
      type: pvc
      claimName: "pvc-example-nfs"
      subPath: "/data/${subject.subjectId}"
      createIfNotExist: true
      permissionType: user
      uid: "${subject.uid}"
      gid: "${subject.uid}"
      startDirectory: "/home/${subject.subjectId}"
```

4 Cleanup

- a Undeploy all applications that are using Persistent Volume. For undeploying MATLAB Online Server, matlab-pool, use the following command:

```
./mosadm undeploy matlab-pool
```

- b Remove Persistent Volume and Persistent Volume Claim:

```
kubectl delete -f
```

- c Uninstall the NFS CSI driver:

```
helm uninstall csi-driver-nfs -n kube-system
```

EBS (Elastic Block Storage) PVC Storage Profile

The following example provides instructions to create a Persistent Volume Claim where the data is stored in an AWS EBS volume, and for using this newly created Persistent Volume Claim in MATLAB Online Server to provide data persistence to the MATLAB user upon signing in.

1 Pre-Requisites

The cluster must install the required EBS CSI Driver. To install the driver, refer to the information at [Amazon EBS CSI driver](#).

You might need to set permissions to allow MATLAB Online Server to create the EBS volumes and attach them to the Kubernetes cluster through IAM roles.

2 Create Persistent Volume Claim

Create Persistent Volume Claim using yaml and kubectl.

- a Save the following lines in a yaml file, for example, "ebs-pv.yaml."

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-example-ebs-eks
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: gp2
  resources:
    requests:
      storage: 100Gi
```

- b** Create Persistent Volume and Persistent Volume Claim using the command:

```
kubectl apply -n mathworks -f ebs-pv.yaml
```

Note that you do not need to create a Persistent Volume explicitly; the GP2 storage class automatically provisions an EBS volume when a Persistent Volume Claim is created.

3 Storage Profile

```
storage:
  profiles:
    - name: home
      mounts:
        - name: home
          mountPath: "/home/${subject.subjectId}"
          type: pvc
          claimName: "pvc-example-ebs-eks"
          subPath: "/data/${subject.subjectId}"
          createIfNotExist: true
          permissionType: user
          uid: "${subject.uid}"
          gid: "${subject.uid}"
          startDirectory: "/home/${subject.subjectId}"
```

4 Cleanup

For more information, see Amazon EBS CSI driver.

EFS (Elastic File System) PVC Storage Profile

The following example provides instructions to create a Persistent Volume Claim where the data is stored in an AWS EFS volume, and for using this newly created Persistent Volume Claim in MATLAB Online Server to provide data persistence to the MATLAB user upon signing in.

1 Pre-Requisites

The cluster must install the required EFS CSI Driver. To install the driver, refer to the information at Amazon EFS CSI driver.

You might need to set permissions to allow MATLAB Online Server to create the EFS volumes and attach them to the Kubernetes cluster through IAM roles.

2 Persistent Volume Claim

Create Persistent Volume Claim using yaml and kubectl.

- a Save the following lines in a YAML file, for example, "efs-pv.yaml."

```
kind: StorageClass
  apiVersion: storage.k8s.io/v1
  metadata:
    name: efs-sc
  provisioner: efs.csi.aws.com
  ---
  apiVersion: v1
  kind: PersistentVolume
  metadata:
    name: pvc-example-efs-eks
  spec:
    capacity:
      storage: 5Gi
    volumeMode: Filesystem
    accessModes:
      - ReadWriteMany
    persistentVolumeReclaimPolicy: Retain
    storageClassName: efs-sc
    csi:
      driver: efs.csi.aws.com
      volumeHandle: <EFS Volume ID > # fs-8416d972
  ---
  apiVersion: v1
  kind: PersistentVolumeClaim
  metadata:
    name: pvc-example-efs-eks
  spec:
    accessModes:
      - ReadWriteMany
    storageClassName: efs-sc
    resources:
      requests:
        storage: 5Gi
```

- b Create Persistent Volume and Persistent Volume Claim using the following command:

```
kubectl -n mathworks apply -f az-files-pv.yaml
```

3 Storage Profile

```
storage:
  profiles:
    - name: home
      mounts:
        - name: home
          mountPath: "/home/${subject.subjectId}"
          type: pvc
          claimName: "pvc-example-efs-eks"
          subPath: "/data/${subject.subjectId}"
          createIfNotExist: true
          permissionType: user
          uid: "${subject.uid}"
          gid: "${subject.uid}"
          startDirectory: "/home/${subject.subjectId}"
```

4 Cleanup

For more information, see Amazon EBS CSI driver.

Azure Files PVC Storage Profile

The following example provides instructions to create a Persistent Volume Claim where the data is stored in an Azure Files volume, and for using this newly created Persistent Volume Claim in MATLAB Online Server to provide data persistence to the MATLAB user upon signing in.

1 Pre-Requisites

The cluster must install the required Azure Files CSI Driver. To install the driver, see Use Azure Files Container Storage Interface (CSI) drivers in Azure Kubernetes Service (AKS).

You must allow permissions for MATLAB Online Server to access the Azure Files volumes and attach them to the Kubernetes cluster. Create a Kubernetes secret with the appropriate credential information (storage account and the storage key):

```
kubectl create secret generic -n mathworks az-files-example-secret --from-literal=azurestorageaccountname=<STORAGEACCT> --from-literal=azurestorageaccountkey=<STORAGEKEY>
```

2 Persistent Volume Claim

Create Persistent Volume Claim using yaml and kubectl.

To provide access to the Azure Files, perform the following steps:

- a** Save the following lines in a YAML file, for example, "az-files-pv.yaml".

```

apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-example-az-files
spec:
  capacity:
    storage: 5Gi
  accessModes:
    - ReadWriteMany
  azureFile:
    secretName: az-files-example-secret
    shareName: <AZ-Files-Share-Name>
    readOnly: false
  mountOptions:
    - dir_mode=0777
    - file_mode=0777
    - uid=1000
    - gid=1000
    - mfsymlinks
    - nobrl
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-example-az-files
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: ""
  resources:
    requests:
      storage: 5Gi

```

- b** Create Persistent Volume and Persistent Volume Claim using the following command:

```
kubectl -n mathworks apply -f az-files-pv.yaml
```

3 Storage Profile

```

storage:
  profiles:
    - name: home
      mounts:
        - name: home
          mountPath: "/home/${subject.subjectId}"
          type: pvc
          claimName: "pvc-example-az-files"
          subPath: "/data/${subject.subjectId}"
          createIfNotExist: true
          permissionType: user
          uid: "${subject.uid}"
          gid: "${subject.uid}"
          startDirectory: "/home/${subject.subjectId}"

```

4 Cleanup

For more information, see [Use Azure Files Container Storage Interface \(CSI\) drivers in Azure Kubernetes Service \(AKS\)](#).

Authorization

By default, as soon as you create the MATLAB Pool, all users are authorized to access its MATLAB instances. To customize which user groups can access this pool, in the `resourceDefinition` section of `matlab-pool.yaml`, update the `allowedGroups` field. For example:

```
resourceDefinition:
  allowedGroups:
    - "team1"
    - "team3*"
```

This sample YAML enables access to any "team1" user and any user whose team name starts with "team3" (team31, team32, team3abc, and so on) by using a wildcard pattern. MATLAB Online Server uses the glob pattern, which is an industry standard for specifying wildcard patterns. For more details, see [https://en.wikipedia.org/wiki/Glob_\(programming\)](https://en.wikipedia.org/wiki/Glob_(programming)).

Session Timeouts

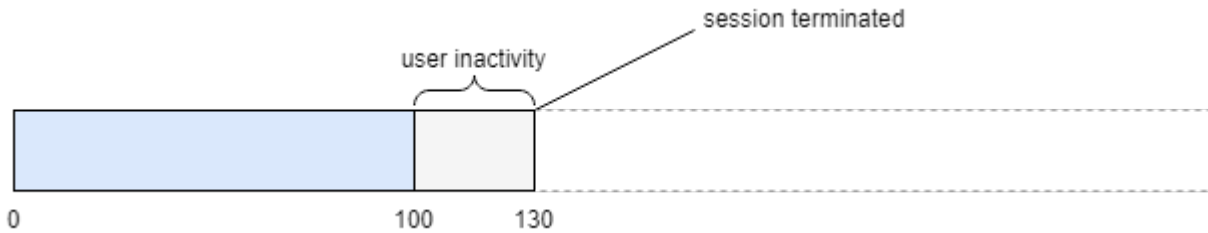
In the `resourceDefinition` section of the `matlab-pool.yaml` file, these fields configure how long before MATLAB sessions time out. The default field values are shown. Units are in minutes.

```
resourceDefinition:
  maxLifetimeMinutes: 240
  defaultLifetimeMinutes: 120
  maxInactivityTimeoutMinutes: 30
  defaultInactivityTimeoutMinutes: 15
```

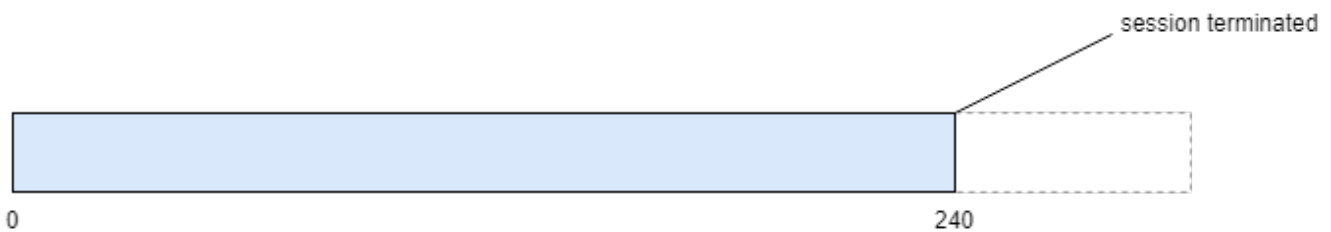
This table describes the type of session timeouts you can configure and which YAML fields to update.

Goal	Field to Update
Set the number of minutes that MATLAB Online Server waits before terminating the MATLAB session.	<code>maxLifetimeMinutes</code>
MATLAB Online Server terminates the session regardless of whether the user is still active.	Note For simplicity, consider setting <code>defaultLifetimeMinutes</code> to the same value as <code>maxLifetimeMinutes</code> .
Set the number of minutes that the MATLAB Online user interface waits for activity before terminating the MATLAB session.	<code>maxInactivityTimeoutMinutes</code>
Session activity is defined as any mouse or keyboard movements or any API requests.	Note For simplicity, consider setting <code>defaultInactivityTimeoutMinutes</code> to the same value as <code>maxInactivityTimeoutMinutes</code> .

This diagram shows a session that terminates due to inactivity. After 100 minutes, the user becomes inactive. After 30 consecutive minutes of inactivity (`maxInactivityTimeoutMinutes` and `defaultInactivityTimeoutMinutes` are set to 30), the MATLAB Online interface terminates the session.



This diagram shows a session that terminates when it reaches the maximum session time. The user remains active for the entire session, but after 240 minutes (`maxLifetimeMinutes = 240`), the MATLAB Online interface terminates the session.



MATLAB Pod Metadata and Environment Variables

Each user who logs into MATLAB is assigned a separate MATLAB pod. In the `resourceDefinition` section of the `matlab-pool.yaml` file, you can specify the following pod metadata:

Pod Metadata	YAML Field to Update
Kubernetes tags	<code>podLabels</code>
Kubernetes annotations	<code>podAnnotations</code>
MATLAB environment variables that are required to run the MATLAB process	<code>env</code>

Use this sample `matlab-pool.yaml` code, and update the field values for your MATLAB pod:

```
resourceDefinition:
  ownerMetadata:
    podLabels:
      - name: username
        value: ${subject.subjectId}
      - name: environment
        value: "prod"
    podAnnotations:
      - name: username
        value: ${subject.subjectId}
    env:
      - name: MOS_USERNAME
        value: ${subject.subjectId}
      - name: environment
        value: "prod"
```

The `${subject.subjectId}` format enables the AuthNZ service to dynamically populate the user metadata when that user signs in. For more details about how placeholders work, see “Placeholders” on page 3-29.

Title Bar

To update the title bar text that displays on the browser tab when a user signs in, update the `displayName` field of `matlab-pool.yaml`. For example:

```
poolConfig:  
  displayName: "R2022b"
```

You can set any value, such as "R2022b 4 CPU, 16 GB".

After the server establishes a MATLAB session, the display name appears as the title of the browser tab in this format:

```
MATLAB Online - R2022b
```

Configure Multiple MATLAB Pools

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 R2021b, R2021b with GPU access, or R2022a.

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.

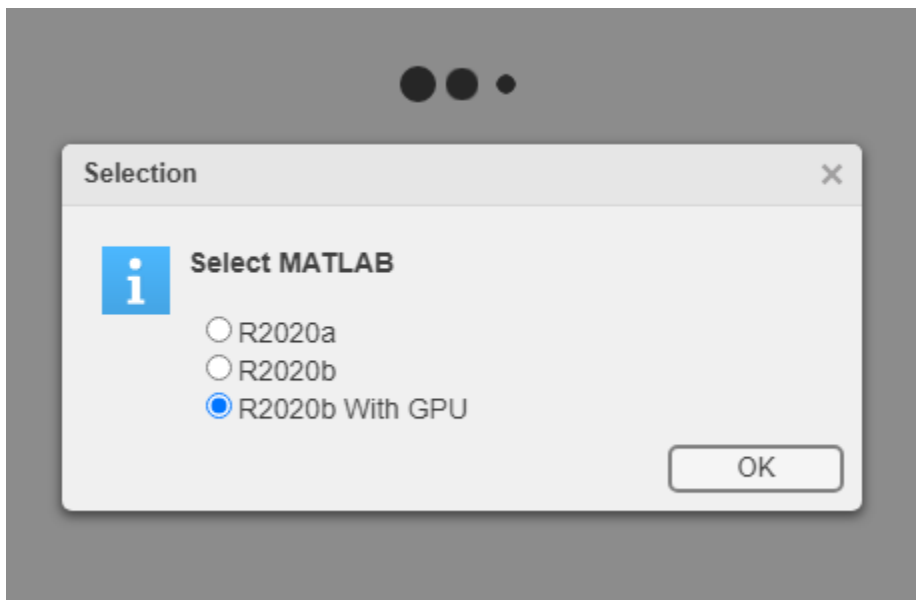
Enable Multiple Pool Chooser

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 “Install Kubernetes” on page 2-3), 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:



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

Create Multiple Pools

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, `r2022a.yaml`, `r2022bgpu.yaml`, `r2022b.yaml`, and so on.
- 4 (Optional) Configure the MATLAB pool (see “Configure MATLAB Pool” on page 3-14).
- 5 Configure additional pool options (see “Configure Additional Pool Options” on page 3-43).

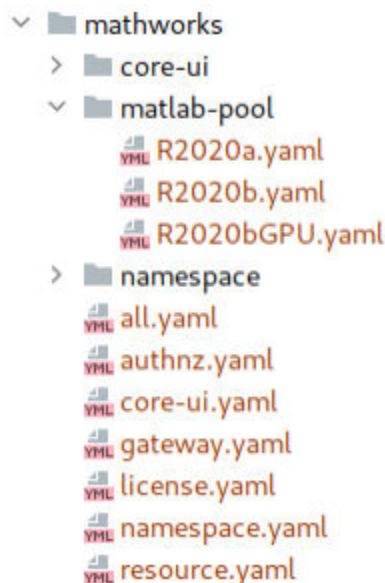
Additional MATLAB Pools

For each MATLAB pool you want to add, follow these steps:

- 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, `r2022a.yaml`) to `matlab-pool/rxxxxy.yaml`, where `xxxxxy` is the release version of the additional MATLAB you installed.
- 3 (Optional) Configure the new MATLAB pool (see “Configure MATLAB Pool” on page 3-14).
- 4 Configure additional pool options for the new MATLAB pool (see “Configure Additional Pool Options” on page 3-43).

After configuring all the individual pools, specify a single version of the MATLAB documentation to use across all MATLAB pools. See “Specify MATLAB Documentation” on page 3-43.

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



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 `rxxxxxy.yaml` file, for example, for `r2022b.yaml`:

```
poolConfig:
  displayName: "R2022b"
```

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

Property	Default Value	Description
<code>displayName</code>	Release name and number, for example, R2022b	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.

Specify MATLAB Documentation

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

- 1 Select the version of MATLAB (for example, R2022b) that you want to serve the help content. Open the respective override (`r2022b.yaml`), and then 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.setDocRelease('R2022b');docRelease=com.mathworks.mlservices.MLHelpServices.getDocRelease;clear docRelease;"
includeHelpSearch: false
```

Replace "R2022b" with the MATLAB version you are using for the help content.

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
```
- 1 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.

Add New Pool to MATLAB Pools Already Running

You can add a new MATLAB pool to a cluster that already has pools deployed and running. After creating the new pool (see “Configure MATLAB Pool” on page 3-14), use this command to deploy it:

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

The `--install` flag enables you to deploy your new MATLAB pool without having to redeploy all running pools, which results in server downtime.

MATLAB Online Server Administration

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.

Manage Docker Images

Load, list, and push Docker images.

Load Docker Image

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

Syntax

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

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
```

List Docker Images

List all the Docker images in the local Docker cache.

Syntax

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

Example Output

The output values depend on the `BASE_IMAGE_FLAVOR` that you specified in the base image container configuration (debian or rhel).

The following example is for a debian base image:

```
mathworks/mos/com.mathworks.matlabonlineserver.matlab-image:R2021a  
mathworks/mos/com.mathworks.matlabonlineserver.matlab-omnibus-debian-  
image:R2021a  
mathworks/mos/com.mathworks.matlabonlineserver.omnibus-debian-image:R2021a  
mathworks/mos/com.mathworks.matlabonlineserver.license-debian-image:R2021a  
mathworks/mos/com.mathworks.matlabonlineserver.core-ui-debian-image:R2021a
```

```
mathworks/mos/com.mathworks.matlabonlineserver.gateway-image:R2021a
mathworks/mos/com.mathworks.matlabonlineserver.resource-image:R2021a
mathworks/mos/com.mathworks.matlabonlineserver.resource-proxy-image:R2021a
```

This next example is for a rhel base image:

```
mathworks/mos/com.mathworks.matlabonlineserver.matlab-image:R2021a
mathworks/mos/com.mathworks.matlabonlineserver.matlab-omnibus-rhel-image:
R2021a
mathworks/mos/com.mathworks.matlabonlineserver.omnibus-rhel-image:R2021a
mathworks/mos/com.mathworks.matlabonlineserver.license-rhel image:R2021a
mathworks/mos/com.mathworks.matlabonlineserver.core-ui-rhel-image:R2021a
mathworks/mos/com.mathworks.matlabonlineserver.gateway-image:R2021a
mathworks/mos/com.mathworks.matlabonlineserver.resource-image:R2021a
mathworks/mos/com.mathworks.matlabonlineserver.resource-proxy-rhel-image:R2021a
```

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.

Syntax

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

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
```

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:R2021a docker-registry.com/repo-
prefix/mos/com.mathworks.matlabonlineserver.matlab-image:R2021a
docker push docker-registry.com/repo-
prefix/mos/com.mathworks.matlabonlineserver.matlab-image:R2021a
```

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

```
...
```

Build MATLAB Image

Build an image containing MATLAB from /MATLAB.

Syntax

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

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
...
```

Deploy, Upgrade, and Undeploy Services

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

Deploy Service

Syntax

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

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
```

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
RESTARTS  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

```

Upgrade Service

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

Syntax

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

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.

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-    1      36m
profiles
==> v1/DaemonSet
NAME                               DESIRED  CURRENT  READY  UP-TO-
DATE  AVAILABLE  NODE SELECTOR  AGE
mathworks-matlab-pool-apparmor-    1        1        <none>
loader 1          1          1          1      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-  36m
ui
==> v1/Pod(related)
NAME                               RESTARTS  AGE  READY  STATUS
mathworks-matlab-pool-749657898b-  0/2      0s   ContainerCreating  0      0s
6hnz5
mathworks-matlab-pool-749657898b-  2/2      36m  Running              0      36m
7x84t
mathworks-matlab-pool-749657898b-  0/2      0s   ContainerCreating  0      0s
fnrw2
mathworks-matlab-pool-749657898b-  2/2      36m  Running              0      36m
vnbs9
mathworks-matlab-pool-apparmor-loader-  1/1      36m  Running              0      36m
29lfb
mathworks-matlab-pool-helpsearch-8547f5bdf4-  1/1      36m  Running              0      36m
58hx6
mathworks-matlab-pool-ui-6c999698d4-  1/1      36m  Running              0      36m
5kln9
==> 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

Undeploy Service

Remove the service from the cluster.

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
```

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
```

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.

Syntax

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

Bootstrap or Reset Single-Node Kubernetes Cluster

Use `mosadm` to bootstrap or to reset the cluster.

Note For Red Hat/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
```

Bootstrap Node

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

Syntax

```
mosadm bootstrap-node
```

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.

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>
- <https://packages.cloud.google.com/apt/dists/kubernetes-xenial/main>

Updates

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

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 (1.18.10)
- kubeadm (1.18.10)
- kubectl (1.18.10)
- helm (3.2.3)

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).

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"
  }
}
```

For more details, refer to the topic [Daemon configuration file](#) on Docker docs for more details.

Miscellaneous

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

Description for Red Hat/CentOS Systems

When bootstrap-node is executed on Red Hat/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.

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)

Updates

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

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 (1.18.10)
- kubeadm (1.18.10)
- kubectl (1.18.10)
- helm (3.2.3)

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 Red Hat distribution, `firewalld` is installed by default).

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"
}
```

For more details, refer to the topic [Daemon configuration file](#) on Docker docs for more details.

Miscellaneous

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

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.

Syntax

```
mosadm reset-node
```

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/docker/shim
```

```
/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.

Configure Overrides

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

Kubernetes Configuration

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.

Syntax

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

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
```

Copy Helm Charts

Copy Helm charts into the install location

Syntax

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

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
```

Generate Overrides

Generate Helm override files based on `install.config`.

Syntax

```
mosadm generate-overrides
```

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/R2021a)
Did not find an installation of MATLAB at the expected location:
/usr/local/MATLAB/R2021a
```

Set up the MATLAB install or MATLAB Docker image, and then try the command again. The output you see should resemble 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:R2021a
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
```

Additional Configuration Options

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

Monitoring

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`.

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`.

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
```

Example Output

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

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

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.

Syntax

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

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

Example Output

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

```
your-namespace-authnz      1    Thu Jan 16 16:51:39 2020    DEPLOYED    authnz-1.0.0      1.0.0      your-namespace
your-namespace-core-ui     1    Thu Jan 16 16:51:40 2020    DEPLOYED    core-ui-1.0.0     1.0.0     your-namespace
your-namespace-gateway     1    Thu Jan 16 16:51:41 2020    DEPLOYED    gateway-1.0.0     1.0.0     your-namespace
your-namespace-license     1    Thu Jan 16 16:51:42 2020    DEPLOYED    license-1.0.0     1.0.0     your-namespace
your-namespace-matlab-pool 1    Thu Jan 16 16:51:43 2020    DEPLOYED    matlab-pool-1.0.0 1.0.0     your-namespace
your-namespace-namespace   1    Thu Jan 16 16:51:45 2020    DEPLOYED    namespace-1.0.0   1.0.0     your-namespace
your-namespace-resource    1    Thu Jan 16 16:51:46 2020    DEPLOYED    resource-1.0.0    1.0.0     your-namespace
nginx-ingress              3    Sat Nov 23 15:21:31 2019    DEPLOYED    nginx-ingress-1.25.0 0.26.1c   nginx-ingress
```

Make sure each service has the value DEPLOYED.

Get Pod Descriptions

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

```
kubectl describe pod podId
```

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`.

Example Output

Pod Descriptors	
Name	<i>your-namespace-matlab-pool-5858d796b9-lqslb</i>
Namespace	<i>your-namespace</i>
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	<pre> 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:R2021a" Normal Pulled 19m kubelet, private-node-3 Successfully pulled image " mathworks.com/matlabonlineserver/com.mathworks.matlabonlineserver. matlab-omnibus-debian-image:R2021a" 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:R2021a" Normal Pulling 19m kubelet, private-node-3 Pulling image " mathworks.com/matlabonlineserver/com.mathworks.matlabonlineserver. omnibus-debian-image:R2021a" Normal Pulled 19m kubelet, private-node-3 Successfully pulled image " mathworks.com/matlabonlineserver/com.mathworks. matlabonlineserver.omnibus-debian-image:R2021a" 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:R2021a" 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:R2021a" Normal Pulled 19m kubelet, private-node-3 Successfully pulled image " mathworks.com/matlabonlineserver/com.mathworks.matlabonlineserver. matlab-omnibus-debian-image:R2021a" Normal Created 19m kubelet, private-node-3 Created container windowmanager Normal Started 19m kubelet, private-node-3 Started container windowmanager </pre>

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.

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
```

Examples

```
your-namespace-core-ui-core-ui-6679d5ff7f-rdtfb_your-namespace_core-  
ui-165cfda8ff5c4d2b01838d0acb573df881e04a08c329afcda1b96b9c15cd0450.log
```

```
your-namespace -gateway-65697586f7-vsvls_your-namespace  
_gateway-78afff6773618381f1fc6ba05e63ecef3f3a6fd5de3defd6f50b28a97bb7a44.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
```

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.

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
```

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
```

Set Up Kubernetes Dashboard for MATLAB Online Server

The Kubernetes Dashboard (also known as Web UI) provides a UI that you can use to monitor your cluster. These instructions show how to install and configure the dashboard for a MATLAB Online Server single-node configuration. You can use these instructions as the basis for a larger deployment and set of security configurations in your environment.

Prerequisites

- You have MATLAB Online Server installed on a single node, as described in “Online Install on Single Machine” on page 2-2.
- The Kubernetes `kubectl` command is configured, as described in “Install Kubernetes” on page 2-3 of the installation instructions.

Install and Set Up Dashboard on Single-Node Configuration

These instructions walk you through the basic installation and setup of the Kubernetes Dashboard. For detailed instructions and security configuration options, in the Kubernetes documentation, see <https://kubernetes.io/docs/tasks/access-application-cluster/web-ui-dashboard>.

- 1 Deploy the Kubernetes Dashboard. On the <https://github.com/kubernetes/dashboard> page, copy the installation command for the latest version of the dashboard and execute it in your MATLAB Online Server terminal. The command has this format:

```
kubectl apply -f ...
```

- 2 Verify that the installation succeeded by checking the list of pods in your namespaces.

```
kubectl get pods --all-namespaces
```

Verify that the list displays two pods in a new `kubernetes-dashboard` namespace. For example:

NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
...					
kubernetes-dashboard	dashboard-metrics-scraper-7c857855d9-7zj45	1/1	Running	0	10m
kubernetes-dashboard	kubernetes-dashboard-6b79449649-lpz7t	1/1	Running	0	10m
...					

- 3 Create a Kubernetes service account in your cluster. This account enables authentication into the dashboard.

To create this account, in your MATLAB Online Server root folder, create a file named `matlab-online-server-admin-account.yaml` and enter this information, exactly as shown:

```
<matlab_online_server_root>/matlab-online-server-admin-account.yaml
```

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: matlab-online-server-admin
  namespace: kube-system
---
apiVersion: rbac.authorization.k8s.io/v1
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
```

Apply the file to your Kubernetes cluster.

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

- Retrieve the bearer token for the created `matlab-online-server-admin` account. Execute this command, and copy or save the returned token. In a later step, you use this token to access the dashboard locally.

```
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}'
```

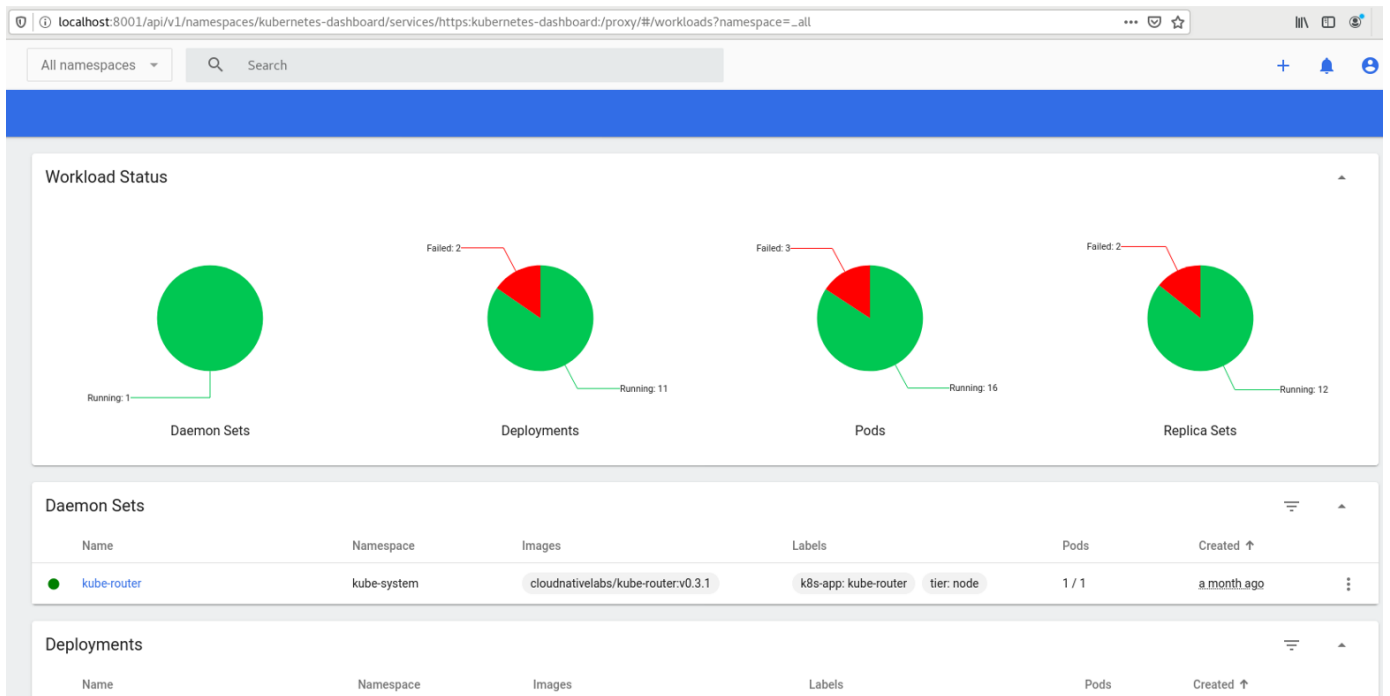
- Start a proxy to enable access to the Kubernetes Dashboard.

```
kubectl proxy
```

- Access the Kubernetes Dashboard on a local browser at this URL:

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

- When prompted for the authentication method, select **Token** and enter the bearer token you retrieved in a previous step.
- Log in to the dashboard. You can now familiarize yourself with the different namespaces for your environment and the administrative features in the Kubernetes Dashboard.



Install and Set Up Dashboard on Managed Kubernetes Provider

The instructions for setting up the Kubernetes Dashboard depends on your provider.

- **Amazon Web Services (AWS®)** — Set up Kubernetes Dashboard using the Elastic Kubernetes Service managed by AWS. In the Amazon documentation, see [//docs.aws.amazon.com/eks/latest/userguide/dashboard-tutorial.html](https://docs.aws.amazon.com/eks/latest/userguide/dashboard-tutorial.html).
- **Microsoft® Azure** — Set up Kubernetes Dashboard using the Azure Kubernetes Service managed by Azure. In the Microsoft documentation, see [//docs.microsoft.com/azure/aks/kubernetes-dashboard](https://docs.microsoft.com/azure/aks/kubernetes-dashboard).
- **Google Kubernetes Engine** — The open-source Kubernetes Dashboard on the Google Kubernetes Engine managed by Google is deprecated. Google recommends using the Google Cloud console instead.

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” on page 5-6.

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  2/2     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.

Option 2: Integrate with Splunk (Advanced)

Follow these instructions to integrate Splunk with MATLAB Online Server. For more information on Splunk, see [Splunk Connect for Kubernetes on EKS!](#)

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.

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.

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.

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.

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 network license manager server.

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
```

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         2/2     Running   0          107s
your-namespace-matlab-pool-8c679f76-cckkp         2/2     Running   0          107s
your-namespace-matlab-pool-ui-579fc8ff49-2bz25    1/1     Running   0          107s
your-namespace-ingress-nginx-controller-kcwwd     1/1     Running   0          104s
your-namespace-resource-resource-894447df7-r9ngb  1/1     Running   0          103s
```

Problems and Resolutions

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

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:

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

You can also search for a license manager error:

```
kubectll 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 network license manager logs). Then, making sure that you use the correct license file, try again.
Cannot connect to license server	<ul style="list-style-type: none"> • 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 the network license manager only on ports 27000 and 27001 by default (controlled by license.dat). Make sure your network license manager 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 then restart the pod: <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <pre>networkPolicy: enabled: true egress: ports: - port: <flexlm_port_here> protocol:</pre> </div>

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:

```
kubectll get pods -n your-namespace
```

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

```
ubuntu@ip-172-31-23-150:~/matlab_online_servers$ kubectll 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-ingress-nginx-controller-5c6b5f775f-7ls87 1/1      Running  0          27h
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 your-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 0x000007f2b5858f8c0 fl::log I Now logging to console stream
00000004 2020-Feb-12 19:46:27.286544 0x000007f2b5858f8c0 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 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 R2021a only.

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,wsz=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 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” on page 7-9.

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” on page 5-14.

- 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.

Pod Creation Stuck

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

- “ImagePullBackOff” on page 5-18
- “CrashLoopBackOff” on page 5-18
- “Pending” on page 5-19
- “ContainerCreating” on page 5-20

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.

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	<p>This means either that the specified registry name is nonexistent, or you do not have access to the container.</p> <p>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.</p> <p>As a test, use the following command to try to pull the same image from your local machine:</p> <pre>docker pull <docker-image-name:docker-image-tag></pre>
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	<p>This means that a specified image cannot be found.</p> <p>To test if the image has an appropriate tag, try to pull the image locally:</p> <pre>docker pull <docker-image-name:docker-image-tag></pre>

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” on page 5-2.

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” on page 5-20 for assistance.

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 0/2
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.

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.

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” on page 5-14.
 - 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.

Evicted or Terminated Pods

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
```

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
```

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
```

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](#) at [kubernetes.io](#).

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
```

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” on page 5-20), and “ContainerCreating” on page 5-20.

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.

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 then 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>
```

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.

Possible Action 1: Provision More Nodes

Provision more nodes on the cluster to meet the resource needs of the cluster.

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 then update with appropriate limit values.
- 2 Run the following mosadm command to update the node:

```
mosadm upgrade matlab-pool
```

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.

License Service Pod State is Running but Users Cannot Connect to MATLAB Online

When the services are running successfully and can connect with the network license manager license server, then the status of pods is Running.

However, if the services are having issues connecting with network license manager, 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          2/2     Running   0           107s
your-namespace-matlab-pool-8c679f76-cckkp          2/2     Running   0           107s
your-namespace-matlab-pool-ui-579fc8ff49-2bz25     1/1     Running   0           107s
your-namespace-ingress-nginx-controller-kcwwd      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.

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.
```

Possible Issues

- Network license manager server listening on ports other than 27000/27001

Network policy is enabled by default for the license service, which allows communicating with network license manager 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 network license manager server is listening on ports other than 27000 and the MLM daemon on other than 27001, update the license service override file and then restart the pod:

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

- The network license manager server is using an incorrect or an expired license file or the network license manager server is misconfigured

Check the network license manager server logs to see details about the reason for failure. Try to connect to the network license manager server from a local machine using the following commands:

```
nc -vz <FlexLM_DNS_NAME> 27000
nc -vz <FlexLM_DNS_NAME> 27001
```

In cases where the network license manager server is using an expired license file:

- Update the license file. See “Update Network License” in the Help Center.
- Set up both a listener on the network license manager 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” on page 5-23.
- See “MATLAB Pod Issues” on page 5-14.

Get Help

If you are unable to resolve the issue with these resources, do the following:

- 1 Have ready the output from the following kubectl commands:

```
kubectl describe pod <license-pod-name> --namespace <namespace>
kubectl logs <license-pod-name> --namespace <namespace>
```

- 2 Contact MathWorks Support.

Fix Security Exception Error When Accessing Help

Problem

Your end user runs a doc command at the MATLAB command prompt, for example:

```
>> doc plot
```

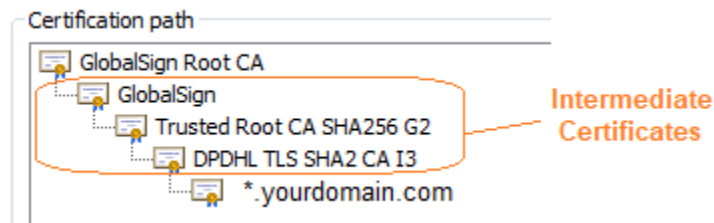
They see an error similar to the following:

```
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
. . .
```

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 contain 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-----
```


Server Maintenance

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
```

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`.

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.

Examples

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 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 Docker
2020/05/28 18:18:22 DockerRootDirSpace : PASS
2020/05/28 18:18:22 Docker : PASS
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 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 CoreUI
2020/05/28 18:18:23 CheckFavoriteIcon : PASS
2020/05/28 18:18:23 CoreUI : PASS
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 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 MATLABPoolUI
2020/05/28 18:18:23 CheckRemoteProxyScript : PASS
2020/05/28 18:18:23 MATLABPoolUI : PASS
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

```

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
DomainBase : mos-test.mathworks.com
2020/05/28 18:20:15 DomainBaseLookup : PASS
2020/05/28 18:20:15 DomainBaseSecurelyReachable : PASS
2020/05/28 18:20:15 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
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

```

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.

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.

Fail

If you get a **FAIL** for one or 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.

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`, `kubectl`, `l` or `helm`) is installed on your machine and properly working.

Properties Tests

The properties checks focus on tests related to your MATLAB Online Server domain.

- `DomainBaseLookup` checks if your domain is reachable securely via https.
- `DomainBaseSecurelyReachable` checks if your domain is resolvable to an IP address.
- `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
```

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" on page 1-6 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.

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 Kubernetes 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.

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" on page 6-2, 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 network license manager 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” on page 5-11), 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, perform the following:

- 1 Save the *logs* folder, generated by the smoke tests, as a zip file.
- 2 Run the command `sudo ./mosadm gather-system-info`. This command generates the file `install.log`.
- 3 Contact [MathWorks Support](#).

Parallel Computing Toolbox and MATLAB Parallel Server - Administrators

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 Parallel 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 Parallel 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.

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
```

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](#) in the Help Center.

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 MATLAB Online Server 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 MathWorks License Manager 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
```


End User Instructions

For the end user instructions, see “Parallel Computing Toolbox and MATLAB Parallel Server—End Users” on page 8-11.

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 [Get Started with the Database Toolbox](#) in the Help Center. 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 page 7-4 on the MathWorks website). In addition, you need the database host and port information so you can properly configure the back end.

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 file. See “Configure Network Policies” on page 7-9.

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
```

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 on page 3-24 to create a read-only mount that contains JDBC drivers for your MATLAB Online Server.

End User Instructions

See “Database Access with MATLAB Online” on page 8-8 for instructions to give the end user for accessing their database.

Product Maintenance

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.

Prerequisites

Before you begin, make sure you have done the following tasks:

- Installed the Linux version of MATLAB from the [Downloads](#) page and launched MATLAB successfully.
- Learned what support packages are and how to manage them on the Linux version of desktop MATLAB.
- Installed and configured MATLAB Online Server.

Install Procedure

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 2022 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
```

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.

Step 3. Install Support Packages

- 1 From within the restarted MATLAB, sign in to your MathWorks account.
- 2 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.

- 3 To verify that the support packages are installed successfully, launch MATLAB and then execute the following command:

```
>> supportPackages = matlabshared.supportpkg.getInstalled
```

- 4 To verify that the support packages are available in MATLAB Online, launch MATLAB in the browser and then execute the following command:

```
>>addons = matlab.addons.installedAddons
```

- 5 Exit MATLAB.
- 6 Delete the licenses folder under /MATLAB.

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 then 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 license service using the following commands:

```
mosadm undeploy license
```

```
mosadm deploy license
```

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.

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 and Manage Add-Ons](#) in the Help Center.
- Familiarize yourself with MATLAB Online Server installation and configuration. See Single Machine Installation on page 2-2.

Install Procedure

Step 1. Create Docker Image with Add-Ons Installed

- 1 Go to the [Downloads](#) at mathworks.com. Install the Linux version of MATLAB R2021a in the folder /MATLAB.
- 2 Install the required add-ons using the Get and Manage Add-Ons instructions on mathworks.com.

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 then 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
```

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 then 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 add-ons: *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.

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 then 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,acregmin=600,acregmax=600,acdirmin=600,acdirmax=600,hard,nocto,noacl,proto=tcp,timeo=600,retrans=2,mountproto=tcp,local_lock=none,nolock"
          restrictNavigationOutsideMounts: false
      addons:
        directory: "/MATLAB/MATLAB Add-Ons"
```

Option 2: Read-Only Add-Ons Directory Through Special Mount

The add-ons you select are installed and then rolled out to the entire organization. MATLAB users cannot install any custom add-ons.

- 1 Create a MATLAB Docker image. See “Prepare Docker Images” on page 2-7 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,wsiz=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,no
lock"
        - name: addons
          mountPath: /AddOns
          type: nfs
          permissionType: none
          server: "nfs"
          subPath: "/exports/matlab/R2021a/addons"
          mountOptions:
"ro,relatime,vers=3,rsize=1048576,wsiz=1048576,namlen=255,acregmin=600,acregmax=600,acdirmin=600,acdirmax=600,hard,nocto,noacl,proto=tcp,timeo=600,retrans=2,sec=sys,mountproto=tcp,local_lock=none,n
olock"
          restrictNavigationOutsideMounts: false
          addons:
            directory: "/AddOns"
```

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 “Prepare Docker Images” on page 2-7 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/${subject.subjectId}"
          mountOptions:
            "rw,relatime,vers=3,rsize=1048576,wsiz=1048576,namlen=255,a
            cregmin=600,acregmax=600,acdirmin=600,acdirmax=600,hard,noct
            o,noacl,proto=tcp,timeo=600,retrans=2,mountproto=tcp,local_l
            ock=none,nolock"
          restrictNavigationOutsideMounts: false
      addons:
        directory: "/home/${subject.subjectId}/MATLAB Add-Ons"
```

Update MATLAB Release

MathWorks may release MATLAB updates following the 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 then install the latest MATLAB update for the release of your MATLAB Online Server installation.
- 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:R2021a
mathworks/mos/com.mathworks.matlabonlineserver.matlab-
image:R2021a-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:R2021a
remote-registry.com /mos/com.mathworks.matlabonlineserver.matlab-
image:R2021a-backup
```

```
docker push remote-registry.com
/mos/com.mathworks.matlabonlineserver.matlab-image:R2021a-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
```

Configure Network Policies

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.

Prerequisites

Before you begin, make sure you have completed the following tasks:

- Install and then configure MATLAB Online Server. See [Single Machine Installation](#) on page 2-2.
- 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 then 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.

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.

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`.

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.

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 “Authentication” on page 3-5.

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 in the Help Center.

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.

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 the MATLAB documentation in the Help Center.

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 “Configure MATLAB Pool” on page 3-14.

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” on page 7-10 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](#).

Override Egress Rules

To override egress rules, see the specification [Network Policy v1 beta1 extensions](#) at [kubernetes.io](#).

This example of a network policy override, defined in the MATLAB Online Server override YAML file, allows the application running inside the particular Kubernetes pod/service (for example, license or matlab-pool) to have full access to port 53 on all nodes/machines.

```
networkPolicy:
  egress:
    - ports:
      # allow dns access
      - port: 53
```

This next example of a network policy override, instead of applying the rule to all pods as in the previous example, is applied to pods that are tagged with the color blue.

```
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](#).

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)

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 MathWorks License Manager 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” on page 5-23 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).

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” on page 3-2.

After consulting the information presented in this section, add the desired configuration and then perform an upgrade (undeploy and deploy) of the service using mosadm (see “Manage MATLAB Online Server” on page 4-2).

Example: MATLAB Communication with MySQL Database

Database details:

- Server: mymysql.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 - mymysql.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 MathWorks License Manager 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” on page 6-12.

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 “Authentication” on page 3-5 for more details.

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 then 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” on page 5-2.

Update MATLAB Online Server

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 on page 2-2.

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.

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).

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**

Step 2. Download MATLAB Online Server Update

- 1 Go to [//www.mathworks.com/products/matlab-online-server.html](http://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 MATLAB Online Server on the new server or new namespace.

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.

Deploy MATLAB Online Server:

- a** Execute the following command to install all 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.

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

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

The pods running MATLAB can take several minutes to fully start (2/2 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” on page 5-14.

- 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).

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” on page 5-2.
- 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” on page 5-14 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
- Mac OS: /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 “Verify MATLAB Online Server Installation” on page 2-9 of the MATLAB Online Server installation instructions.

MATLAB Online Server in an HA Environment

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

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.

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” on page 3-19.

High Availability

Because its main services start as single pods, MATLAB Online Server is not highly available (HA) by default. To enable HA for MATLAB Online Server, in the file `install.config`, set the flag `IS_HA_ENABLED` to `true`. This setting ensures that the generated overrides have a replica count set to 2 for each MATLAB Online Server service.

```
IS_HA_ENABLED=true
```

While most of MATLAB Online Server services are stateless, two services (AuthNZ and Gateway) store, in memory by default, information about logged in users and the mapping between user sessions and their respective MATLAB instances.

If the flag `IS_HA_ENABLED` is set to `true`, then MATLAB Online Server starts 4 pods of a Redis™ service that works as a highly available cache in the cluster. In this case, a failure of any AuthNZ or Gateway pods does not impact the stored information about current sessions and their mapped tokens and MATLAB instances. Note that MATLAB Online Server Pods are configured to be spread across several nodes using anti-affinity in the case where the Kubernetes cluster has multiple worker nodes.

Although HA can be enabled for MATLAB Online Server regardless of the underlying Kubernetes infrastructure, you should have a highly available Kubernetes cluster for your production setup. Refer to your Kubernetes provider documentation on how to create a highly available cluster.

Deployment under Red Hat OpenShift

If you are deploying MATLAB Online Server under Red Hat OpenShift, then you need to add the following lines to the generated `./overrides/<cluster>/<namespace>/redis.yaml` file:

```
serviceAccount:
  create: false
  name: custom-sa
```

The field `custom-sa` is a service account that is created following the instructions for installing MATLAB Online Server on Red Hat OpenShift. Those instructions are:

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

External Redis Server

While setting the flag `IS_HA_ENABLED` to `true` is enough for many customers, you may prefer to use an external Redis server instead. In this case, in the file `install.config`, you must set an additional flag, `DEPLOY_REDIS`, to `false` to disable the deployment of Redis locally.

```
IS_HA_ENABLED=true
DEPLOY_REDIS=false
```

In addition, you must configure the external Redis server manually in two override files:

- `./overrides/<cluster>/<namespace>/authnz.yaml`
- `./overrides/<cluster>/<namespace>/gateway.yaml`

By default, MATLAB Online Server generates a Redis configuration with the type `FailoverRedis` in those two files, as follows:

```
cache:
  type: FailoverRedis
  failoverredis:
    mastername: redis-sentinel-master
    addrs: mathworks-redis:26379
```

This configuration is for communicating with the Redis deployed within MATLAB Online Server which is of type `FailoverRedis` and its specific configuration is under the field `failoverredis`.

Depending on your external Redis configuration, you must change the type of the cache and its specific configuration. In addition to the FailoverRedis type, MATLAB Online Server supports two additional types: BasicRedis and ClusterRedis.

While BasicRedis refers to a single instance of Redis where all items are stored in the same instance, ClusterRedis refers to multiple instances of Redis where items are sharded across all instances and each instance has only a subset of items.

Note that not all Redis configurations provided by cloud platforms support HA by default. If you are opting for an external Redis, use one with an HA configuration enabled. For secure communication with any external Redis server, set the `tls` field to `true`.

BasicRedis Configuration

```
cache:
  type: BasicRedis
  basicredis:
    addr: your-redis-server:your-redis-port
    tls: true
    password: your-redis-password
```

ClusterRedis Configuration

```
cache:
  type: ClusterRedis
  clusterredis:
    addrs: your-redis-server:your-redis-port
    tls: true
    password: your-redis-password
```

The `addrs` field, under the ClusterRedis type, can be assigned to a single `server:port` value or to multiple comma “,” separated `servers-i:port-i` values.

MATLAB Online Server End User Guides

MATLAB Online Using MATLAB Online Server

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.

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.

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	<p>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.</p> <p>The following commands can help you adjust your code to a Linux platform:</p> <p><code>fullfile</code> builds full file names from component parts with the appropriate file separator characters.</p> <p><code>filesep</code> returns the platform specific file separator character.</p> <p><code>fileparts</code> gets parts of file name.</p> <p><code>path</code> lets you view or change search path.</p> <p><code>pathsep</code> searches path separator for current platform.</p>
File Management	<ul style="list-style-type: none"> • Upload <p>To upload files within MATLAB Online, use the Upload button on the toolbar or drag and drop your file into the Current Folder browser.</p> • Download <p>To download a file, use the toolbar button or the command line. You must have access to the file.</p>
MEX Command	The MEX command, for building C/C++ or Fortran MEX files, is not supported.

System

Specification	Value
Operating system	Linux
MATLAB version	R2021a
Compute resources	Check with your administrator for exact resources. The recommendation is a minimum of 4 cores and 8 GBs per user.

Support browsers	<p>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®.</p> <p>For the best overall experience, Google Chrome™ is recommended.</p>
------------------	--

MATLAB Sessions

Operation	Description
Session Timeout	<p>The default idle time for MATLAB Online is 15 minutes.</p> <p>Idle time is determined by inactivity within the user interface. Your administrator can choose to increase the idle time duration.</p>
Session Save and Restore	<p>The following items are automatically saved and restored between MATLAB Online sessions:</p> <ul style="list-style-type: none"> • MATLAB path • MATLAB settings and preferences • Files in your home folder <p>The following items are not automatically saved and restored between sessions:</p> <ul style="list-style-type: none"> • Workspace variables • Dynamic Java® classpath

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” on page 7-2 and “Install Add-Ons for MATLAB Online” on page 7-4 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	<ul style="list-style-type: none"> • aeroDataPackage • Aero.FlightGearAnimation • Aero.VirtualRealityAnimation
Communications Toolbox™	RF propagation functionality

Computer Vision Toolbox™	<ul style="list-style-type: none"> • 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™	<ul style="list-style-type: none"> • 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™	<ul style="list-style-type: none"> • GPU Coder app • Configuration parameter dialog box • coder.screener • MATLAB Coder™ Support Package for NVIDIA® Jetson® and NVIDIA DRIVE® Platforms • Deep Learning code generation for TensorRT targets
Image Processing Toolbox™	<ul style="list-style-type: none"> • implay • immovie • Image Segmenter • Image Region Analyzer app • Image Batch Processor app
Mapping Toolbox™	

MATLAB Coder	<ul style="list-style-type: none"> • 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™	<ul style="list-style-type: none"> • Wavelet Analyzer app • Wavelet Signal Denoiser app • Signal Multiresolution Analyzer app

Functionality

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+/

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.

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 (see [Webcam Support in MATLAB Online](#)).
- It can communicate with Raspberry Pi hardware (see [Connect to Raspberry Pi Hardware Board in MATLAB Online](#)).

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.

Database Access with MATLAB Online

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 mathworks.com.
 - 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**. In the Help Center, see [What Is the MATLAB Search Path?](#), 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](#).

The following example MATLAB 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](#) in the Help Center.
- 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 to 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.

Set Up GitHub for Use with MATLAB Online Server

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.
-

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')
```

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
```

Parallel Computing Toolbox and MATLAB Parallel Server—End Users

MATLAB Online Server includes support for Parallel Computing Toolbox and the use of MATLAB Parallel Server Clusters (also known as MATLAB Parallel Server in some documentation) from the MATLAB Online Server user interface command window.

You can access the `parpool` functionality (in Parallel Computing Toolbox) from the command window. The different ways of using this functionality are illustrated in the following sections.

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.

Use MATLAB Parallel Server Cluster

To use a MATLAB Parallel Server Cluster, the MATLAB Online Server user must upload the profile for the cluster before using it. This profile is created by the MATLAB Parallel 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 in the Help Center:

- [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 Parallel Server.

Parallel Computing Toolbox Disabled by MATLAB Online Server Administration

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.
>>
```


MATLAB Online Server Supporting Resources

Resources and Links

AuthNZ

Authentication and authorization. See <https://en.wikipedia.org/wiki/Authentication>.

Docker

<https://www.docker.com/>

FlexNet Publisher (formerly FLEXlm)

The MathWorks License Manager uses this technology to check for licenses.

[//www.flexerasoftware.com/monetize/products/flexnet-licensing.html](http://www.flexerasoftware.com/monetize/products/flexnet-licensing.html)

Helm Chart

<https://helm.sh>

Kubernetes

<https://kubernetes.io/>

NGINX

<https://en.wikipedia.org/wiki/Nginx>

PEM

PEM format requirements for certificates and domain keys

PEM, DER, CRT, and CER: X.509 Encodings and Conversions

SAML

<https://wiki.oasis-open.org/security/>

Transport Layer Security (TLS)

https://en.wikipedia.org/wiki/Transport_Layer_Security

YAML

<https://yaml.org/>