

System requirements

September 10, 2026

System requirements

PAMS deployment requirements describe the frontend and API resources used to prepare a deployment. Open Home after signing in.

If you need to...	Go to
Check the resources required before deployment	Before you start
Locate the files that define deployment	Find the deployment definitions
Understand the frontend and API baseline	Understand the deployment baseline
Prepare the frontend	Prepare the frontend deployment
Prepare the API	Prepare the API deployment
Compare development and production	Handle development and production deployments
Check the application entry point	Check the application entry point
Continue with the application	Dashboard overview
Get help with an unresolved requirement	Getting help

Before you start

Before you prepare a deployment, collect the storage, node, CPU, and memory values below. The

API uses two `azurefile` volumes: `10Gi` for `App_Data` and `5Gi` for `Data`. The API pods target Windows nodes with `agentpool: win2`.

Component	Request	Limit
API	<code>250m</code> CPU, <code>512Mi</code> memory	<code>500m</code> CPU, <code>1Gi</code> memory
Frontend	<code>100m</code> CPU, <code>128Mi</code> memory	<code>500m</code> CPU, <code>512Mi</code> memory

When these values are ready, open the four deployment definitions.

Find the deployment definitions

Use the four deployment definitions as your checklist.

Prerequisites:

- You can open the four deployment definitions.
- You know whether you are preparing the development or production definition.

Steps:

1. Open `Frontend/Dockerfile` to check the frontend build image, serving image, and exposed port.
2. Open `Pams/API/Pams.API/Dockerfile` to check the API base image and file permissions.
3. Open `pams-frontend-deployment.yaml` to check the frontend deployment, service, and ingress definitions.
4. Open `pams-api-deployment.yaml` to check the API volumes, deployment, service, and ingress definitions.

Result: You have the four deployment definitions needed to prepare the frontend and API.

Understand the deployment baseline

The frontend is a static web application built with Node 16 and served by NGINX. The API

runs on the ASP.NET 4.8 Windows Server Core LTSC 2022 image. The API stores application data

in `App_Data` and `Data` ; both the frontend and API use port 80 for their service endpoints.

Area	Frontend	API
Build or base image	Node 16 build image	ASP.NET 4.8 Windows Server Core LTSC 2022
Serving	NGINX serves <code>/app/dist/pams</code>	IIS serves the API image
Storage	No volume is declared in the frontend deployment	<code>App_Data</code> and <code>Data</code> volume mounts
Traffic handler	NGINX frontend service	IIS API service

Area	Frontend	API
Entry routing	Frontend ingress	API ingress

The API image creates `C:\Data` and grants access to the IIS worker identities. The API

deployment uses the Windows node, CPU, and memory values listed in [Before you start](#).

After you understand this baseline, prepare the frontend deployment.

Prepare the frontend deployment

Prepare the frontend image and its deployment settings.

Prerequisites:

- The frontend build context contains `package*.json`, the application source, and `nginx.conf`.
- The frontend deployment has the resource values from [Before you start](#) available.

Steps:

1. In `Frontend/Dockerfile`, locate `BUILD_ENV` and record its default value, `development`.
2. In `Frontend/Dockerfile`, locate `npm run build:${BUILD_ENV}` and record the build command.
3. In `Frontend/Dockerfile`, locate the frontend build output and its NGINX destination.
4. In `Frontend/Dockerfile`, locate `EXPOSE 80` and record the frontend container port.
5. In `pams-frontend-deployment.yaml`, locate the frontend CPU and memory requests and limits and compare them with [Before you start](#).
6. In `pams-frontend-deployment.yaml`, locate the frontend service port and target port.
7. In `pams-frontend-deployment.yaml`, locate the ingress backend port and compare it with the frontend service port.

Result: You have checked the frontend build, NGINX destination, container port, resource, service, and ingress settings.

After the frontend settings are ready, prepare the API deployment.

Prepare the API deployment

Prepare the API image and its deployment settings.

Prerequisites:

- A Windows node matching `agentpool: win2` is available.
- The two `azurefile` volume claims are available with `10Gi` for `App_Data` and `5Gi` for `Data`.

Steps:

1. In `Pams/API/Pams.API/Dockerfile`, locate the ASP.NET 4.8 Windows Server Core LTSC 2022 base image.
2. In `Pams/API/Pams.API/Dockerfile`, locate the command that creates `C:\Data`.
3. In `Pams/API/Pams.API/Dockerfile`, locate the permissions granted to the IIS worker identities for `C:\Data`.
4. In `pams-api-deployment.yaml`, locate the volume mount for `C:\inetpub\wwwroot\App_Data`.
5. In `pams-api-deployment.yaml`, locate the volume mount for `C:\inetpub\wwwroot\Data`.
6. In `pams-api-deployment.yaml`, confirm that the API deployment targets Windows nodes with `agentpool: win2`.
7. In `pams-api-deployment.yaml`, locate the API CPU and memory requests and limits and compare them with [Before you start](#).

Result: You have checked the API image, permissions, storage, node, CPU, and memory settings.

After the API settings are ready, select the environment definition.

Handle development and production deployments

Development and production use separate API deployment definitions. Both environments use the

same storage sizes, node placement, and API resource values; the image names differ.

Prerequisites:

- You have selected the environment definition to apply.
- You have the API storage claims and Windows node available.

Steps:

1. Choose the environment definition that matches your target:

If you are preparing...	Use
Development	The development API deployment and its development image
Production	The production API deployment and its production image

2. Apply the selected API storage, node, and resource values.
3. Compare the selected image and volume claim names with the matching environment block.

Result: The selected environment uses its own API image and deployment definitions while retaining the documented storage, node, and resource settings.

Use [Getting help](#) when a deployment question remains after you apply these settings.

When a requirement is missing

Use the values in the deployment tables for preparation. Obtain the following requirements before you proceed if your installation needs them:

Requirement	Action
Browser support	Obtain the supported browser value before deployment.
Workstation requirements	Obtain the end-user workstation value before deployment.
Database	Obtain the database engine and sizing values before deployment.
Network and version requirements	Obtain the required network and version values before deployment.
Operator dependencies	Obtain the dependency checklist before deployment.

Use the confirmed values when you prepare the deployment.

After these requirements are available, check the application entry point.

Check the application entry point

The frontend service and ingress route traffic to port `80` . After signing in, Home opens the authenticated dashboard surface.

Prerequisites:

- The frontend service and ingress are available on port `80` .
- You know where to select Home in the application navigation.

Steps:

1. Select Home in the application navigation.
2. Check that Home shows `Sort by Booking` , `Sort by Target` , and `Sort by Achievement` .
3. Check that no required marker appears on those three controls.
4. Check that Home shows no data table or action buttons in this view.

Result: Home opens with the three sorting controls visible and no data table or action buttons in this view.