

Oracle® MicroTx

Using MicroTx on Oracle Cloud Marketplace



Release 26.1

G58506-01

July 2026

ORACLE®

Oracle MicroTx Using MicroTx on Oracle Cloud Marketplace, Release 26.1

G58506-01

Copyright © 2026, Oracle and/or its affiliates.

Primary Author: Sylaja Kannan

Contributing Authors: Tulika Das

Contributors: Brijesh Kumar Deo, Bharath MC, Shivanshu Singh

Contents

1	Set Up the Environment	
1.1	Minimum Resource Requirements	1
1.2	Prerequisites	2
2	Deploy Stack	
2.1	Provision MicroTx from Oracle Cloud Marketplace	1
2.2	Access MicroTx Console	3
2.3	View Logs	3
2.4	Delete the Stack (Optional)	4

1

Set Up the Environment

Use Oracle Cloud Marketplace Terraform stack to deploy Transaction Manager for Microservices (MicroTx) on Oracle Cloud Infrastructure (OCI).

The Terraform stack provisions the required OCI infrastructure, prepares the MicroTx container images, deploys MicroTx on Oracle Kubernetes Engine (OKE), and provides deployment information as Terraform outputs.

- [Minimum Resource Requirements](#)
Confirm that your Oracle Cloud tenancy and target region have sufficient capacity for the required resources before launching the stack.
- [Prerequisites](#)
Ensure that your OCI environment meets the deployment requirements.

1.1 Minimum Resource Requirements

Confirm that your Oracle Cloud tenancy and target region have sufficient capacity for the required resources before launching the stack.

Ensure that the capacity is enough for the maximum worker node count, and not just the initial node count. The deployment can fail if sufficient compute, networking, or storage resources are unavailable.

Table 1-1 Minimum Resource Requirements

Resource	Minimum or default requirement
OKE Worker node shape	VM.Standard.E3.Flex, VM.Standard.E4.Flex, or VM.Standard.E5.Flex
OKE Worker node OCPUs	6 OCPUs
OKE Worker node memory	32 GB
OKE Worker boot volume	150 GB
OKE Worker node count	1–10 (autoscaling enabled by default)

Table 1-1 (Cont.) Minimum Resource Requirements

Resource	Minimum or default requirement
Autonomous AI Database database instance (if you want stack to deploy it)	Select the type of Autonomous AI Database instance that you want the stack to provision. If you have logged in to OCI with an Oracle Cloud Free Tier account, you can only provision an Always Free Autonomous AI Database instance. If you have a paid account, you can provision Autonomous AI Database for Developers or Paid instance based on your resource requirements.
<i>Note</i> An instance of Autonomous AI Database for Developers uses 4 ECPUs, while the Always Free and Paid options use 2 ECPUs. All instances require a minimum of 20 GB memory.	
Load balancer	One public load balancer
File Storage	One file system and mount target
Jump host	This is a temporary instance provisioned by the stack during deployment. It is automatically deleted by the stack on the successful completion of Terraform deployment. It requires 2 OCPUs and 32 GB memory.

For additional sizing recommendations, see [MicroTx Installation and Configuration Guide](#).

1.2 Prerequisites

Ensure that your OCI environment meets the deployment requirements.

- An Oracle Cloud account:
 - Sign up for an Oracle Cloud account: [How Do I Sign Up?](#)
 - Sign in for the first time: [Sign In to Oracle Cloud For the First Time](#)
 - Sign in to Oracle Cloud: [Oracle Cloud Account](#)
- Access to an assigned Oracle Cloud tenancy. You must create a compartment in Oracle Cloud Infrastructure (OCI) in your assigned tenancy where the stack can create resources. See:
 - [Setting Up Your Tenancy](#)
 - [Managing Compartments](#)
- Ensure that the stack is available in your OCI region.
- Verify that you have permissions to create OCI networking, Compute, Kubernetes, storage, IAM, and optional Autonomous Database resources in the compartment.
- Obtain an Oracle Cloud Infrastructure Registry (OCIR) user name and authentication token. See [Logging in to Oracle Cloud Infrastructure Registry](#).
- Ensure that your tenancy has sufficient service limits for all required resources, such as file storage, load balancers, and OKE worker nodes. See [Minimum Resource Requirements](#).

- Run the following command to generate a 32-byte encryption key file for MicroTx Workflows.

```
openssl rand -hex 32 > encryption.key
```

The command creates an encryption key file named `encryption.key`. Note the name and location of this file as you will provide it later.

- Decide whether you want the stack to deploy a new Autonomous AI Database instance or if you want to connect to an existing instance of Oracle Database 19c or later.
- To use an existing instance of Oracle Database 19c or later:
 - Grant privileges to the database user to create and run stored procedures. See [Grant Privileges to Database User](#)
 - Upload the wallet ZIP file, if required, to Object Storage in OCI and create a read-only pre-authenticated request (PAR) URL. See [Creating an Object Storage Pre-Authenticated Request](#). Note down the PAR URL as you will need to provide it later while creating the stack.

2

Deploy Stack

Before deploying the stack, ensure that you complete all the prerequisites.

Topics

- [Provision MicroTx from Oracle Cloud Marketplace](#)
- [Access MicroTx Console](#)
- [View Logs](#)
If deployment does not complete successfully, review logs to identify and troubleshoot issues.
- [Delete the Stack \(Optional\)](#)
To remove the deployment, destroy the Terraform stack in OCI Resource Manager.

2.1 Provision MicroTx from Oracle Cloud Marketplace

Before you begin, complete all the prerequisites.

To provision MicroTx from Oracle Cloud Marketplace:

1. Sign in to [Oracle Cloud Marketplace](#).
2. From the Oracle Cloud Marketplace home page, use the search box under Applications and search for the keywords **MicroTx**.
3. From the search results, select MicroTx Free Edition.
4. Click **Launch Stack**.
5. On the Launch Stack page, provide the following information:
 - a. **Compartment**: Select a compartment where you want to build the resources.
 - b. **Version**: Select the version of the stack.
 - c. Review the terms and conditions, and then select the **I have reviewed and accept the Publisher terms and conditions** check box.
 - d. Click **Launch Stack** to launch the stack in the OCI environment.The Create Stack page is displayed. Optionally customize the **Stack Information** fields. All fields are completed with defaults.
6. Under MicroTx Configurations, provide the following information.
 - a. **MicroTx Version**: Select the MicroTx version to deploy.
 - b. **Deployment Name**: Enter a unique deployment name. The name must be less than 32 characters and can contain lowercase letters, numbers, and hyphens. All resources generated by the stack in OCI are prefixed with the specified name.
7. Under Database Configuration, provide information based on whether you want the stack to provision a new Oracle Autonomous AI Database instance or if you want to connect to an existing database.
 - To permit the stack to provision a new Oracle Autonomous AI Database instance:

- a. Select the **Provision Oracle ATP database 26ai instance on OCI** check box.
 - b. Select the type of Autonomous AI Database instance that you want the stack to provision. If you have logged in to OCI with an Oracle Cloud Free Tier account, you can only select **Always Free Autonomous AI Database**. If you have a paid account, you can select **Autonomous AI Database for Developers** or based on your resource requirements. An instance of Autonomous AI Database for Developers uses 4 ECPUs, while the Always Free and Paid options use 2 ECPUs.
 - c. **Database User:** Enter the name of the database user that you want the stack to create. Optionally, use only `_` as a special character in the user name. The stack creates the database user with privileges to access the MicroTx database schema. You cannot change the database username after it is created.
 - d. **Database Password:** Enter the password for the specified database user.
- To connect to an existing instance of Oracle Database 19c or later:
 - **Database User:** Enter the name of the database user with privileges to access the MicroTx database schema.
 - **Database Password:** Enter the password for the specified database user.
 - **Database JDBC URL:** Enter the connection string to the database instance in the following format: `jdbc:oracle:thin:@//<publicIP>:<portNumber>/<database unique name>.<host domain name>`.
 - **Database Wallet PAR URL:** Enter the pre-authenticated request URL to access the database wallet that you have uploaded to Object Storage in OCI as a prerequisite.
8. Under Workflow Server Secret, browse for the `encryption.key` file for MicroTx Workflows that you had created earlier.
9. Under OCIR Configurations, provide the following information.
 - **OCIR Username:** Enter the user name to sign in to Oracle Cloud Infrastructure Registry. See [Logging in to Oracle Cloud Infrastructure Registry](#).
 - **OCIR Auth Token:** The authentication token used to access OCIR. See [Generating an Auth Token to Enable Login to Oracle Cloud Infrastructure Registry](#).
10. Under Kubernetes Cluster Configurations, provide the following information.
 - a. **Node Pool Name:** Name of the node pool for the worker nodes. Use lowercase letters, numbers, and single hyphens only.
 - b. **Enable Cluster Autoscaler for Node Pool:** Select this to enable node pools to autoscale based on resource usage. This setting permits addition and removal of worker nodes as needed.
 - c. **Minimum Number of Worker Nodes:** The minimum number of nodes in the node pool. If autoscaling has not been enabled, this is the total number of worker nodes available. The default value is 1.
 - d. **Maximum Number of Worker Nodes:** The maximum number of nodes in the node pool. This is only available if autoscaling has been enabled. The default value is 10.
 - e. **Worker Nodes Instance Shape:** Select an appropriate compute shape: `VM.Standard.E3.Flex`, `VM.Standard.E4.Flex`, or `VM.Standard.E5.Flex`.
 - f. **Worker Node OCPU Count:** The number of OCPUs for each worker node compute instance. The minimum is 6 OCPUs.
 - g. **Worker Node Memory (GB):** The amount of memory for each worker node compute instance. The minimum is 32 GB.

11. Click **Next**.

The Review page displays all the information that you have provided.

12. On the Review page, review the information you provided and click **Create**.

The **Run Apply** check box is selected by default. This runs the stack and sets up MicroTx.

The Job Details page in Oracle Resource Manager is displayed. An Apply job is started to provision the stack.

13. Periodically monitor the progress of the Apply job until it is finished.

After you apply the Terraform stack, Resource Manager performs the following tasks:

- a. Creates the OCI networking resources.
- b. Creates the OKE cluster.
- c. Creates the worker node pool.
- d. Creates File Storage resources.
- e. Creates the optional Autonomous AI Database.
- f. Uploads deployment files to the temporary jump host.
- g. Downloads the Oracle MicroTx software.
- h. Pushes Oracle MicroTx container images to OCIR.
- i. Installs Istio.
- j. Deploys Oracle MicroTx by using Helm.
- k. Waits for the Kubernetes pods to become ready.
- l. Returns deployment outputs.

The deployment can take several minutes, especially while OKE node pools, load balancers, Autonomous AI Database instance, and container images are being created.

If the job fails, click the job name to view logs.

2.2 Access MicroTx Console

1. From the stack's Job Details page of the successful apply job, under **Resources**, click **Outputs**.
2. Copy the value of `microtx_url`.
3. Enter the MicroTx console URL in any browser.

MicroTx console is supported on browsers that support Oracle JavaScript Extension Toolkit (Oracle JET). For list of supported browsers, see <https://www.oracle.com/webfolder/technetwork/jet-900/globalSupport-FAQ.html#platforms>.

2.3 View Logs

If deployment does not complete successfully, review logs to identify and troubleshoot issues.

- View console logs for the Apply job in Resource Manager. See [Getting Logs for a Job](#).
- View logs for the temporary jump host, if it is still available, at `/var/log/microtx-stack/`. Important log files include:
 - `bootstrap.log`: Logs created during initialization of the temporary jump host.

- `prepare-images.log`: Logs created during MicroTx image pull and local tagging.
- `tagnpush.log`: Logs created during OCIR tagging and push.
- `oke_deployment.log`: Logs created during OKE, Istio, Oracle Cloud Infrastructure File Storage service, Oracle Database, Helm, and pod readiness checks.

If image downloads fail because of temporary DNS or connectivity issues to Oracle Cloud Infrastructure File Storage service, the deployment automatically retries the operation before reporting an error.

2.4 Delete the Stack (Optional)

To remove the deployment, destroy the Terraform stack in OCI Resource Manager.

During the destroy operation, Resource Manager removes:

- Oracle MicroTx Helm release
- Kubernetes namespace
- File storage persistent volumes
- Istio ingress gateway
- OCI infrastructure resources

Ensure that you have deleted the resources before you delete the stack.

Note

Do not manually delete Terraform-managed resources unless the destroy operation cannot complete and you have reviewed the Resource Manager job logs.

Deletion of an Autonomous Database can continue after the Resource Manager job finishes because database deletion is asynchronous.

To delete the stack in OCI Resource Manager:

1. In the Oracle Cloud Infrastructure console, click the navigation menu , select **Developer Services**. Under the **Resource Manager** group, click **Stacks**.
2. Select the name of the stack that you want to delete.
3. Click **Terraform Actions**, and then select **Destroy**.
4. When prompted for confirmation, click **Destroy**.
5. Periodically monitor the progress of the Destroy job until it is finished. Ensure that all resources of the stack are deleted successfully.

If an email address is associated with your user profile, you will receive an email notification.
6. Click **Delete Stack**.