SedonaOffice - Octopus Tentacle & SedonaCloud Install

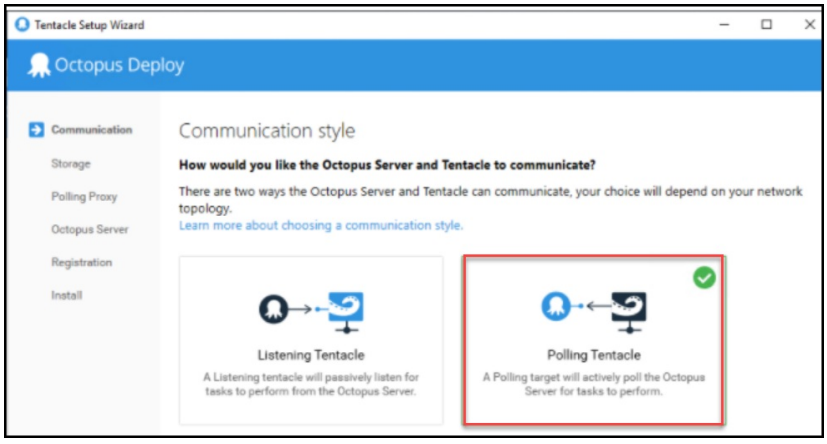
04/09/2025 7:44 am EDT

Required files for IIS Server for SedonaCloud 2.0:

1. Install ASP.NET Core 2.1 Runtime (v2.1.30)
   https://dotnet.microsoft.com/en-us/download/dotnet/thank-you/runtime-aspnetcore-2.1.30-windows-hosting-bundle-installer
2. Install Redis for windows:
   https://github.com/MicrosoftArchive/redis/releases/download/win-3.2.100/Redis-x64-3.2.100.msi
3. Download Tentacle from octopus.com/downloads and install (just click next).

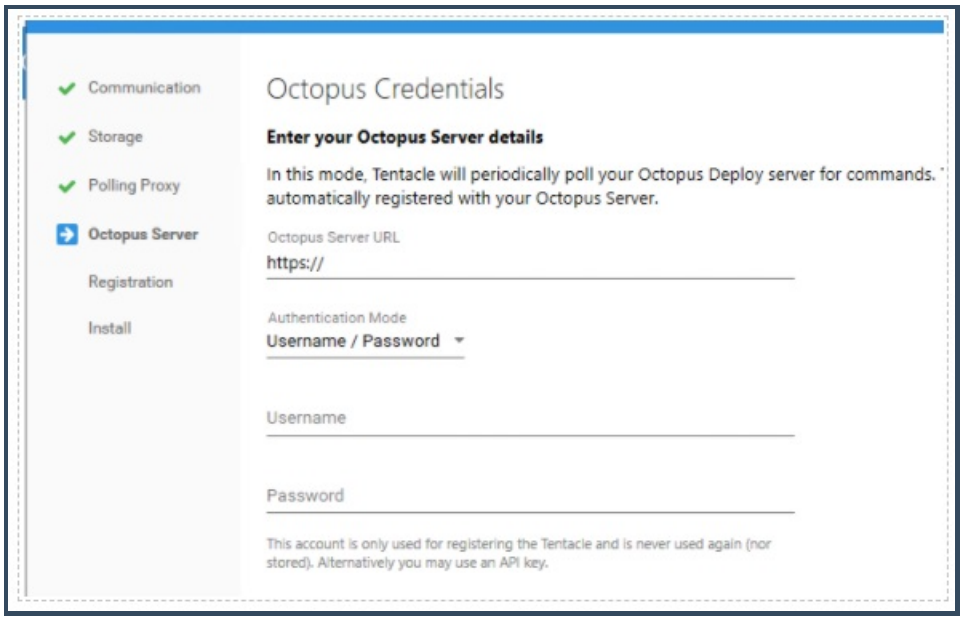
Tentacle Manager will open once installed to configure the tentacle, click Get Started.

Select Polling Tentacle:



Click Next until on the Octopus Server/Credentials screen.

Use created UN/PW and



When you click next, it will validate the credentials entered. Now you register the machine with an Octopus environment and role:

For environment:

- SedonaAPI = API 1.0
- SedonaCloud = API 2.0

For role:

- Give it an identifiable role name. This is used later to identify the deployment target. If the company already has a role (from an API 1.0 deployment), typically we add SC for SedonaCloud (e.g. CustomerSC).

SedonaCloud Install

Access in a browser with the same credentials used to set up Tentacle in a browser.

Find it's easier to find deployment projects by selecting "Projects" tab:

Scroll down to SedonaCloud section.

To start, we will be cloning another Project.

Template Enterprise for Enterprise customers, Template Datacenter for Detroit customers, Template AWS for AWS customers.

After selecting the project to close, go to settings:

Click the 3 dots next to “Save” and select “Clone”:

Name the new project (typically ‘Customer Name’, but ‘Customer Name SedonaCloud’ if upgrading from API 1.0) and click save.

Now you will be editing the newly cloned project.

Select Process on the left and click on “Deploy to Customer” in the center. You should now see where you’re configuring the IIS settings within Octopus.

Change the Step Name to match the company the deployment is for.

For Enterprise or Dedicated Cloud customers: Next to “Execution Plan” click the x on the role (hover over it to get the x) and then enter the role you gave the Octopus tentacle.

For Cloud Customers: Verify the role says “SEDONAAPIASP” if in Det. Datacenter or “SEDONAAPIAWS” if in AWS

For all customers: Verify the Package says “from feed Octopus Server (built-in)”

https://boldwiki.boldgroup.com/link/752#bkmrk-for-cloud-customers%3A-0

For Cloud Customers: You should have a “Custom Install Directory” line. If you do not, click on Configure Features at the top to add:

The install path should look like this:
C:\Octopus\Applications\SedonaCloud\SedonaCloud\SedonaCloud\CustomerName

Enterprise customers should not have the “Custom Install Directory” line unless they need to have multiple API 2.0 endpoints on one server for some reason. One endpoint can serve multiple companies out of the same SedonaMaster, so this should almost never be needed.

If using a custom install path, make sure the checkbox for “Purge this directory before installation” is checked.

For Cloud customers:

Make sure Web Site and ApplicationPool are named CompanyNameSedonaCloud. Enterprise can stay as just SedonaCloud.

For Application Pool:

The best way to set this up is running under the default “Application Pool Identity” for the Identity.

Older deployments may be using a local user that was granted SQL access for the Identity here. Moving forward, it’s

easier to maintain (no PW to expire) to use the default and make some changes I'll note later in the process.

Bindings

For enterprise refer to the preinstall and/or the placeholder site they should have created to verify DNS.

For cloud: You should only need to change the host URL, but also verify the SSL thumbprint is the latest SedonaASP cert.

Once all of that is done, click Save.

Go to Variables (on the left):

Config:CookieDomain – this is the URL without https:// preceding

Config:SedonaCloudClientSecret – generate a GUID (tick the "uppcase" box): https://www.guidgenerator.com/online-guid-generator.aspx

Config:UseSubdomains – Verify this is included. Typically this will be set to "false" now unless the customer wants to use a wildcard SSL cert only for the API and use subdomain endpoints.

ConnectionStrings:DistributedCache – For all customers: This is for the API database (named SedonaCloud). For cloud customers in AWS this is their named instance (full name eg. SQLAWS1\ExampleSQL). For Detroit Datacenter: This is the customer's app server.

ConnetionStrings:SedonaCloudContext – See above.

ConnectionStrings:SedonaDocumentContext – This is the connectionstring to the SedonaDocs DB. For enterprise this is typically their SQL server, but they may have a separate Docs server. This should be noted on the preinstall. For cloud customers this is the named instance in AWS or application server otherwise.

ConnectionStrings:SedonaMasterContext – This is the connectionstring to the SedonaMaster database. This will be the customer's SQL server for enterprise or app server for cloud.

ConnectionStrings:SedonaOfficeContext – Connectionstring for the company database. Cloud customers will either be the communal datacenter server they're on (eg. SQLMain or SCSQL) or the named instance if in AWS.

ConnectionStrings:TempData – as with ConnectionStrings:DistributedCache

License – See LastPass for the link, signature, and keys. On the license generator site:

ExpiryDate: set to year 9999

Name: Customer company name

Email: email on ticket

Max Company: 100

SedonaWeb: check customer systems in SedonaOffice. Check/uncheck accordingly

Click create and copy the generated license into the Octopus Project

LicensePublicKey – Should remain the same, but also in the License generator LastPass note

Urls:IdentityServerUrl and Urls:url – this should be the full host URL including https:// and the port if not using standard 443 (e.g. if an ent customer wants to use port 4443: https://exampleapi.customerdomain.com:4443)

After entering all of that, click “Save”. Now we are ready to package everything as a release and deploy.

Select Releases from the lefthand menu

Click Create Release:

Here we make sure to select the search radio button under Packages to ensure it’s not a pre-release build (there’s a checkbox to hide pre-release builds; they will also have –pre-QA after the build number) and make the Version # match the package version #:

NOTE: If you need to change something in the process or variables and re-release the same version (troubleshooting, SSL cert update, etc.), you will need to make a new release with a new version number. You can use a hyphen to add a note to the end of the version name for what you changed. E.g. 1.14.1.1-SSLthumbprint. This is intended for naming beta builds, but it works better than incrementing the actual build number in cases where we need to redeploy to fix a config mistake or expired SSL/PW.

Click “save”. Then “Deploy to” and select SedonaCloud.

If all goes well, in 1-2 minutes you’ll have the new API endpoint deployed.

We just need to make two quick changes on the IIS server now:

In IIS Manager, go to Application Pools and select the SedonaCloud Application Pool (or CustomerNameSedonaCloud if a cloud customer).

Advanced Settings: Scroll to “Load User Profile” set to True.

Go to C:\Octopus\Applications and open the Properties for the SedonaCloud folder. Make sure IIS_IUSRS has full permission to the folder.

Test it by going to the URL and logging in with the host login:

Select Company from the top menu, then Add Company:

Make sure the company list is populating from SedonaMaster.

That’s it! Provide the customer the host login, URL, SedonaCloud documentation link or pdf, the ClientSecret you

configured and the ClientId which is always SedonaCloudClient.

Common install issues

For the method described above using the default app pool identity, SedonaUser will need at least dbcreator access. If this is not the case, it's easy to diagnose even if not on the SQL server as there will be a Windows event log for the site failing to load that will reference SQL if you look in Event Viewer. If you see that, you know you need to get access to SQL to at least temporarily elevate SedonaUser.

It's also really easy to forget to change "Load User Profile" to true in the advanced app pool settings.

Otherwise, usually, it only fails to start if there is either a missing pre-req (or newer version of .NET core; we use 2.0.9) or the server is not resolving the URL from the Octopus variables to itself. Remember if not using port 443, the Urls: variables need to include the port.

Required files for IIS Server for SedonaCloud 2.0:

Install .Net Core Hosting Bundle 2.0 (.Net Core Runtime) - https://dotnet.microsoft.com/download/thank-you/dotnet-runtime-2.0.9-windows-hosting-bundle-installer
Install Redis for windows: https://dotnet.microsoft.com/download/thank-you/dotnet-runtime-2.0.9-windows-hosting-bundle-installer