

SecureChat: New AppServer Installation and Chat Setup

05/11/2026 2:46 pm EDT

This is for Internal systems and beta customers only. Production setup will differ.

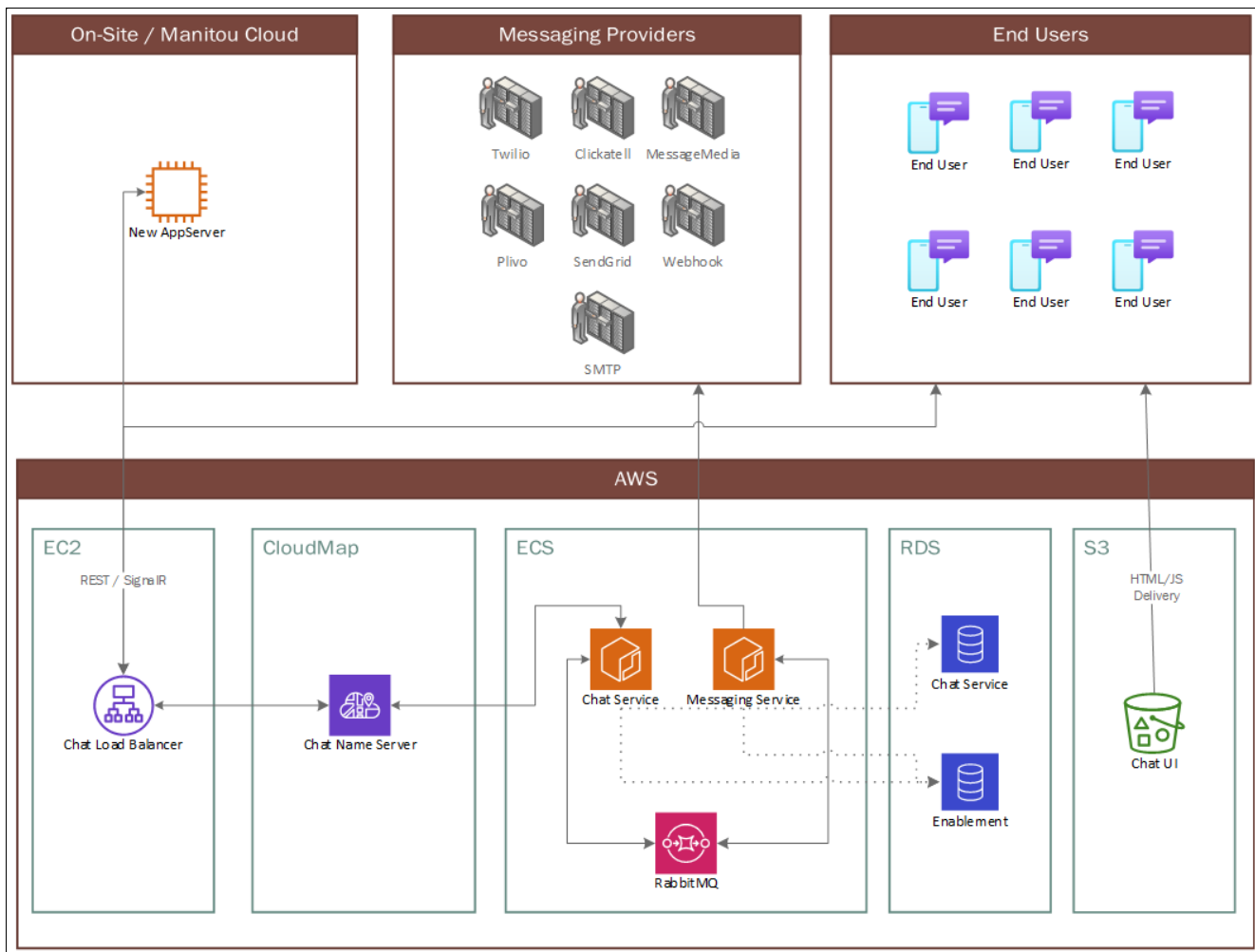
Introduction

This document serves as a guide to setting up an internal machine or setting up a beta customer site to use the new chat feature in Manitou. To make use of the chat feature, the new .NET AppServer is a requirement. However, since the new AppServer is brand new and not yet production tested, we are configuring systems to retain the old AppServer as well. This provides a safe temporary fallback in case issues arise with the new AppServer. The new AppServer is fully backward compatible with the old one, so switching between the two has no adverse effects.

Deployment Diagram

Since most of the deployment is in AWS, this means that deployment to an internal system or beta site is just restricted to the new AppServer. Below is the full deployment diagram for reference.

Note that the AWS deployment is already complete and shared by all customers. Therefore, this document only covers the Manitou side of the deployment.



Prerequisites

Verify that Manitou is on the latest version (v4 as of 03-12-26) — the latest build of the AppServer will be provided by Yanna/team.

.NET is up to date (.zip file provided with all three) — Run as Administrator.

This is the link to .NET 9: <https://dotnet.microsoft.com/en-us/download/dotnet/9.0>

A zip with the .NET 9 installers is attached: [dotnet-runtime-9.0.15-win-x64.zip](#) @

Transfer to New Machine

All files hosted at this link <https://securityandalarminternalkb.knowledgeowl.com/help/files-for-beta-appserver-install>.

AppServer Installation

Since the new AppServer is a folder of many files, it does not currently work well with the distributor. For internal use and for the beta customers, this is going to be a manual file move. **Remember to perform this on every server that is currently running an AppServer at the site.**

Step 1. Copy Files

The new .NET AppServer is being distributed as a ZIP file with all files in it.

Create a target folder for the .NET AppServer at C:\Program Files (x86)\Bold Technologies\Manitou\AppServer.

Extract the files from the .NET AppServer zip to this location.

Once extracted, move the files that are in the “a” folder directly into the AppServer folder.

After extracting the ZIP file, Unblock the files so they can be executed.

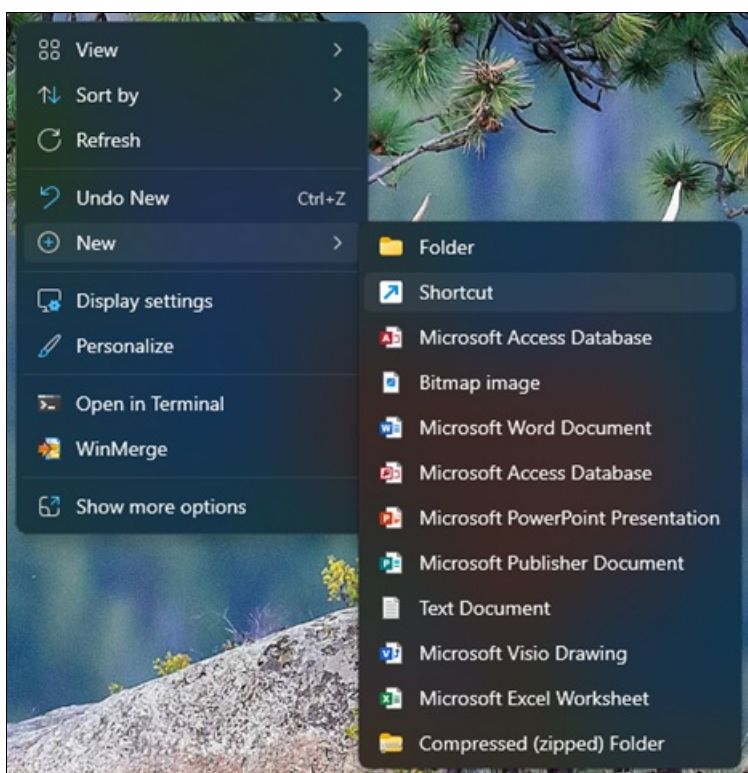
Step 2. Copy Switcher Scripts

There are three PowerShell scripts provided that make switching between the two versions of the AppServer easier:

- old_appserver.ps1 — Switches from the new AppServer to the old AppServer
- new_appserver.ps1 — Switches from the old AppServer to the new AppServer
- which_appserver.ps1 — Reports which AppServer is currently in use

Copy these scripts to the Bold Technologies folder. Then, create shortcuts to the scripts on the desktop using these steps:

1. Right-click on the desktop, choose **New** from the context menu, then choose **Shortcut**.

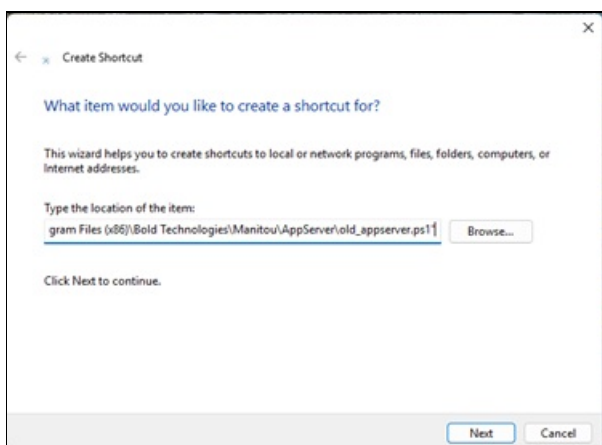


2. In the **Create Shortcut** field, copy the following text for each of the scripts (old_appserver, new_appserver, which_appserver):

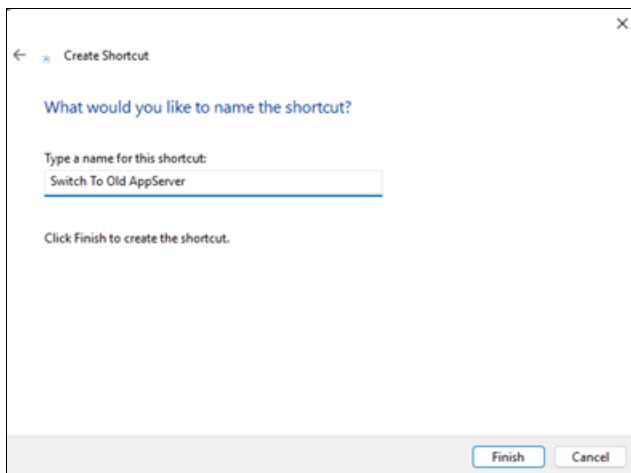
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -ExecutionPolicy Bypass -File "C:\Program Files (x86)\Bold Technologies\old_appserver.ps1"

C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -ExecutionPolicy Bypass -File "C:\Program Files (x86)\Bold Technologies\new_appserver.ps1"

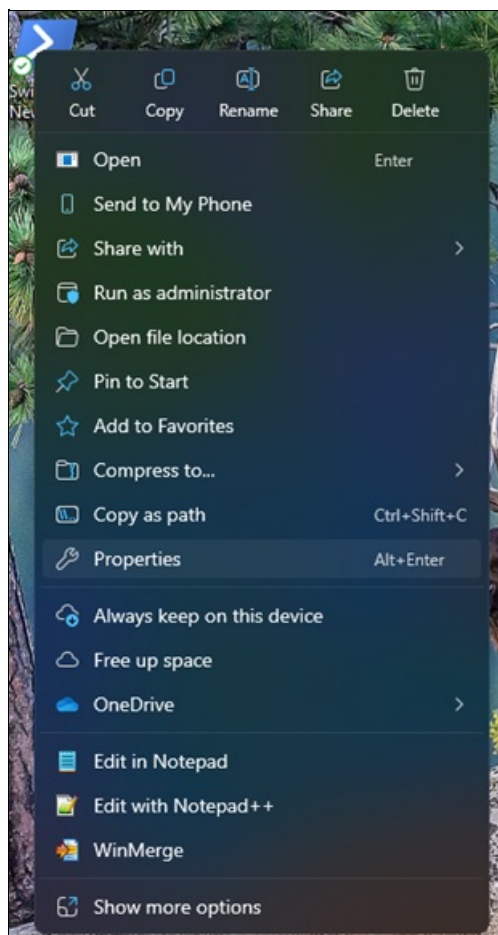
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -ExecutionPolicy Bypass -File "C:\Program Files (x86)\Bold Technologies\which_appserver.ps1"



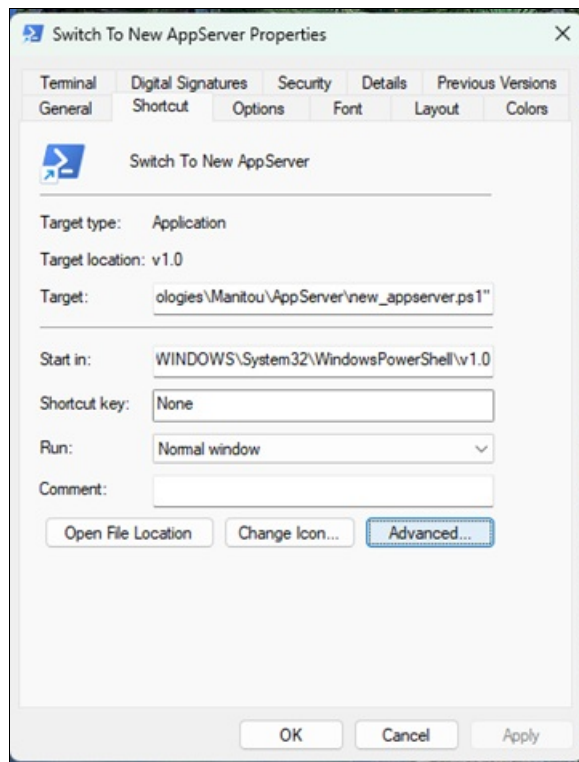
3. Click Next. In the name field, type a descriptive name for the shortcut. In this sample, we created the shortcut for old_appserver.ps1, so we used Switch To Old AppServer:



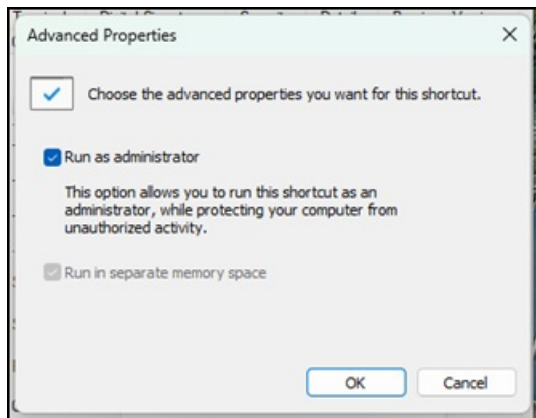
4. Click **Finish**.
5. Repeat this process for the other two scripts.
6. The old_appserver.ps1 and new_appserver.ps1 scripts require users to run them as Administrator, so we need to add UAC escalation to them. For each of those two scripts, follow these steps:
 - a. Right click the shortcut and choose **Properties** from the context menu.



- b. Select the **Shortcut** tab and click the **Advanced...** button.



- c. In the Advanced Properties dialog, select the **Run as administrator** checkbox and click **OK**.



7. Make a copy of all three shortcuts and put in the New AppServer target folder (C:\Program Files (x86)\Bold Technologies\Manitou\AppServer), so they can be utilized from there in the event the desktop icons are not available.
8. Open the Log Viewer and have Application Server enabled with Log Level – Debug (4).
9. Open IIS (Internet Information Services).

Step 3. Switch to new AppServer

1. Right-click on the Application Server in the MSM and select **Stop**.
2. Double-click on the Switch to New AppServer shortcut just created to ensure the system is running on the new AppServer. (The Application Server in the MSM will show as black once you have run the Switch script.)
3. Click on the refresh button on the MSM and make sure that the Application Server now shows red.

4. On IIS, click on the Server connection (in this example, it is BG-MAN-APP01), then click on Restart to restart the IIS service.
5. Now, go back to the MSM, right-click on the Application Server, and click on **Start**.
6. At this point, watch the Log Viewer to confirm that all of the Stored Procedure updates are committed and the tables are cached.

Note that this script will wait for you to press a key before continuing. Use with caution on live production systems.

Configure Manitou for Chat

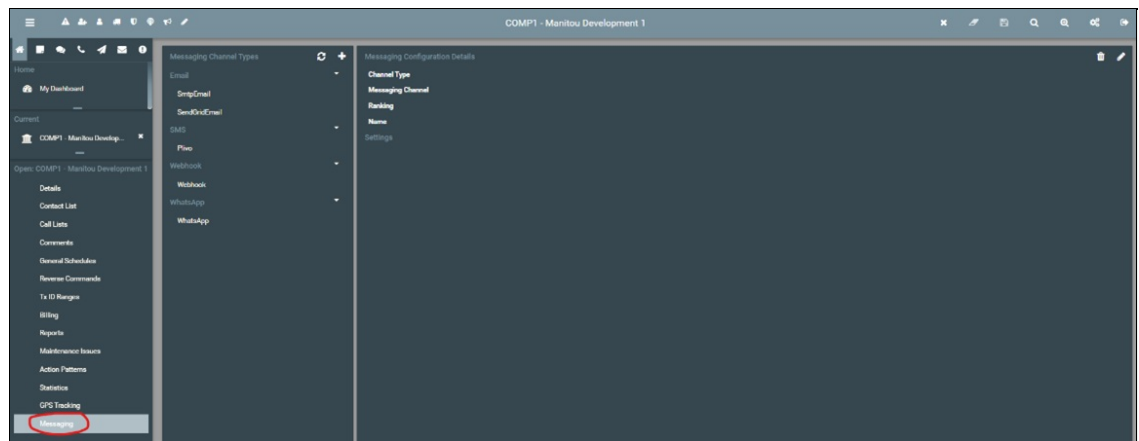
1. Set the URL to the ChatService:
 - a. URL is <https://chat.boldgroup.io>
 - b. Note this URL is on the sandbox/testing server and will change before production release.
 - c. This information goes in the new AppServer's appsettings.json file like this:

```
{  
  "BoldTenantId": "[REDACTED]3e",  
  "BoldTenantSecret": "[REDACTED]1b",  
  "ChatServiceAddress": "https://chat.boldgroup.io"  
}
```

2. Get a Client ID and Client Secret and set them:
 - a. David or Adam can provide this.
 - b. This information goes in the new AppServer's appsettings.json file like this:

```
{  
  "BoldTenantId": "[REDACTED]3e",  
  "BoldTenantSecret": "[REDACTED]1b",  
  "ChatServiceAddress": "https://chat.boldgroup.io"  
}
```

3. Configure the SMS provider account credentials.
 - a. The Customer must have an account on a supported provider and have cleared all regulatory hurdles to enabling sending of SMS messages.
 - b. In the Manitou Web Client, on the Dealer and Monitoring Company pages, there's a Messaging page where you can setup how the dealer and/or central station wants to send chat invitations. When sending an invitation on a customer, Manitou will look for this config on its Dealer first and if not found then will look for the config at the Monitoring Company level.
 - Load the Monitoring Company and switch to Messaging.



- Add new Messaging Channel(s) for the type of chat(s) you want to send.

Messaging Configuration

Channel Type

SMS

Messaging Channel

PlivoSms

Ranking

1

Name

Plivo

The Auth ID to use for sending SMS messages

Auth ID

12

The Auth Token to use for sending SMS messages

Auth Token

Nz

The number the SMS should be from

Source Number

+17

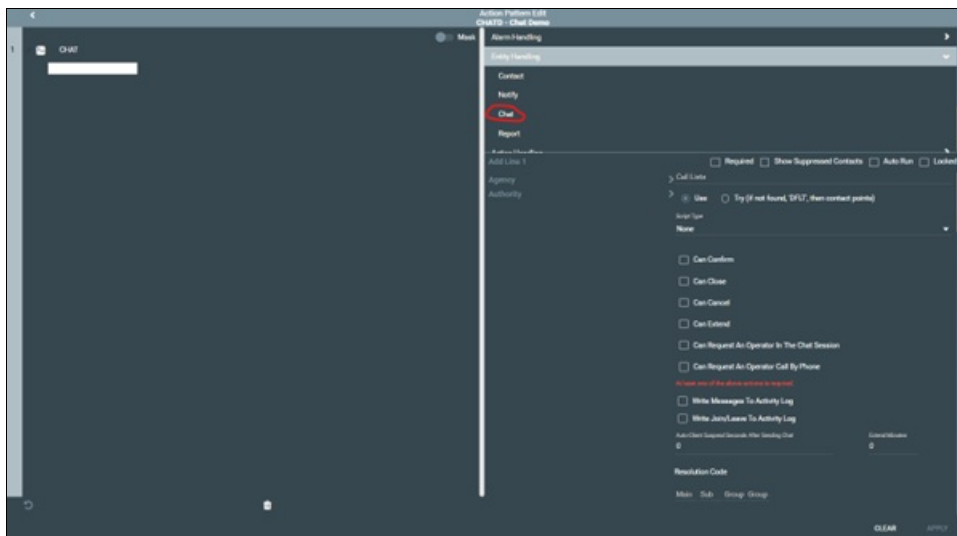
CANCEL

DONE

- If you have more than one Messaging Channel for each type (Email, SMS...) the Ranking field will be used to pick the most appropriate one. The lower the number, the higher the ranking.

Configure an Action Pattern to Use Chat

1. Create/edit an action pattern at any level where you can create action patterns.
2. Add a new action pattern command by going into Entity Handling->Chat.



3. Person Type, Contact Point Type, Call List and Script Type – these are all the same as when using them in a Contact or Notify action pattern command.
4. The “Can” checkboxes govern what participants can do within a chat. All “Can” checkboxes write to the activity log that the action was requested.
 - a. **Can Confirm:** gives participants the ability to confirm an alarm.
 - b. **Can Close:** gives participants the ability to close an alarm. This cancels then closes the alarm. The chat UI will require a password from the participant and Manitou will only cancel/close the alarm if the password that was verified has the Can Cancel Alarm permission. The alarm will be closed with the Resolution Code set in the action pattern command.
 - c. **Can Cancel:** gives participants the ability to cancel an alarm. The chat UI will require a password from the participant and Manitou will only cancel the alarm if the password that was verified has the Can Cancel Alarm permission.
 - d. **Can Extend:** gives participants the ability to extend a schedule for the alarm associated with the chat. The chat UI will require a password from the participant and Manitou will only extend the schedule if the password that was verified has the Can Authorize a Schedule Change permission. The schedule will be extended the number of minutes specified in the Extend Minutes field in the action pattern command.
 - e. **Can Request An Operator In The Chat Session:** gives participants the ability to request an operator join the chat session. If an operator is handing the alarm when the request is made, then a pop-up will notify the operator that it has been requested. The operator can then join the chat if they want to. It does not force the operator to join.
 - f. **Can Request An Operator Call By Phone:** gives participants the ability to request an operator call the chat participant. If an operator is handing the alarm when the request is made, then a pop-up will notify the operator that it has been requested. The operator can then call the participant if they want to. It does not force the operator to call.
5. The **Write** checkboxes govern how the chat reports back to Manitou.
 - a. Write Messages To Activity Log – this will write all messages in the chat to the activity log.
 - b. Write Join/Leave To Activity Log – this will write to the activity log every time a participant joins or leaves the chat.
6. Auto-Client Suspend Seconds After Sending Chat – specifies the number of seconds the auto-client should suspend the alarm after sending the chat.
7. Extend Minutes – specifies the number of minutes a schedule should be extended if a participant extends the schedule.

8. Resolution Code — specifies the resolution code(s) to use if a participant closes the alarm.

Communication Diagram

This diagram is included for reference only. This is the sequence of communication during the lifetime of a chat session.

