Orchestral.ai

Conquer Complexity in Enterprise IT

Getting Started

With Composer

Table of Content

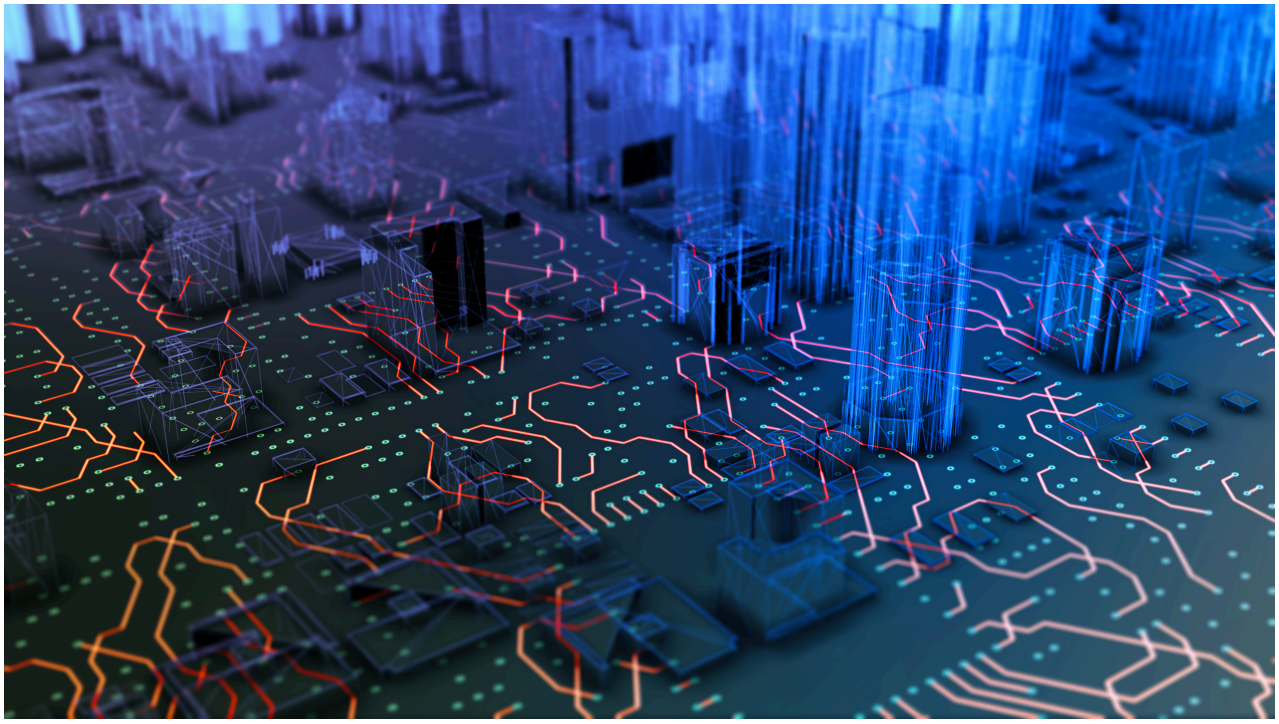
1. Purpose of this document

The purpose of this document is to provide prerequisites for Composer, directions for a fresh install of Composer, simple definitions that are common to Composer and to provide a basic overview of the Composer user interface. If you have any further questions, or for additional help, visit the official Orchestral.ai website (https://orchestral.ai/) and use our contact tab to get ahold of our Client Development Team.

2. Prerequisites

System Requirements	Minimum	Production
Cores	Dual Core CPU	Quad Core CPU
Ram	>2GB RAM	>16GB RAM
Strorage	>10GB Storage	>40GB Storage
Recommended EC2	m5.large	m5.xlarge

Table 2-A. System Requirements



3. Installation of Composer

3.1 Composer Standard Installation

3.1.1 Operating System Supported

Linus (64-bit)	Amazon AWS
Ubunto 18.04 Bionic	Dual Core CPU
RHEL 8 and CentOS 8 RHEL 7 and CentOS 7	Red Hat Enterprise Linus (RHEL) 8 (HVM)

Table 3-A. Supported OS for Standard Install

3.1.2 Pre-Requisites for Installation

1. Internet access to download software packages and repositories
2. Procure license for Composer from Orchestral.ai
    1. Customers should login to Orchestral.ai Client Success Center (CSC) to procure | license file
    2. Non-Customers can obtain a 30 days trail license by browsing on the Orchestral.ai site and submitting the free trial from
3. sudo access to be granted on the server
4. curl installed on the server

To start the installation, please make sure that the operating system you are using meets the above requirements and is at the latest patch level, or a patch level compliant to your enterprises version of either Ubuntu 18 or CentOS/RHEL 8/Oracle Linux 8.

For both Ubuntu and CentOS/RHEL/OL you will need to open the ports 80 and 443 to ensure the Web UI works after the installation of Composer.

Below are the commands to do this utilizing a fresh installation. Including the command to list the ports to ensure they have been added.

For Ubuntu:

```
sudo ufw enable
sudo ufw allow ssh
sudo ufw allow 80,443/tcp sudo ufw status
```

For CentOS/RHEL:

```
sudo firewall-cmd --zone=public --permanent --add-port=80/tcp
sudo firewall-cmd --zone=public --permanent --add-port=443/tcp
sudo firewall-cmd --reload
sudo firewall-cmd --zone=public --list-ports
```

3.1.3 Installation

3.1.3.1 License Enablement

Create a directory “/etc/st2” and place the “license” file in the directory. The “/etc/st2/license” file should be available for the installation to proceed.

Note: The license file does not have an extension associated with the file, it is a standalone file named “license”

3.1.3.2 Install Composer

The username and password you provide in the install script are the name and password you will use to log in to Composer.

Now run the Install Script with your provided License, making sure to enter a username, password and path to license file.

```
curl -sSL https://pkg.orchestral.ai/public/installers/raw/versions/latest/orch-st2-install.sh | bash -s -- -- user=<USER> --password=<PASSWORD> --license=/path/to/license/file
```

Note: This is a command on single line, with curl output being piped into bash
USER, PASSWORD = username and password and license file path.
For example, if you decide username is myUser and password is myPassword then the command is as follows:

```
curl -sSL https://pkg.orchestral.ai/public/installers/raw/versions/latest/orch-st2-install.sh | bash -s -- -- user=myUser --password=myPassword --license=/etc/st2/license
```

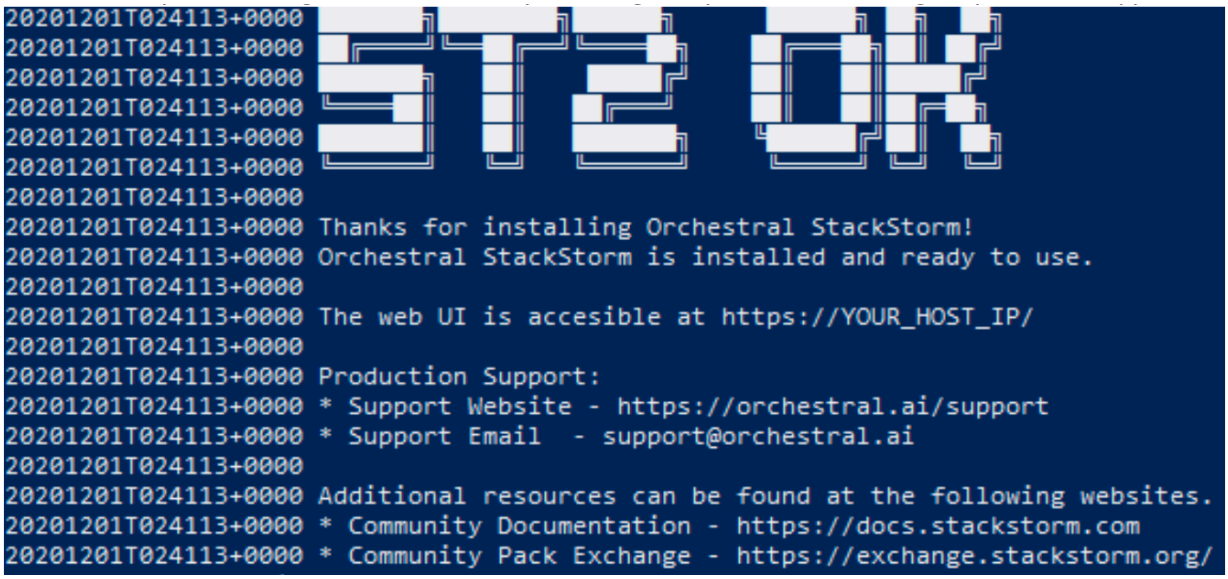
Username “myUser” with password “myPassword” will be created by the script.

Note: If you use shell specific special characters then you should backslash them. For example password myPassword$2345 will be interpreted by the shell as myPassword345. Because $2 is interpreted as second argument, which you don’t have; therefore $2 will be ignored.

For additional documentation, visit https://orchestral.ai/

4. Composer Verification

Once you have run the script, the installation process should be over in as little as 3-4 minutes, depending on the power of the machine you are using. After successfully installing Composer the following output should appear:

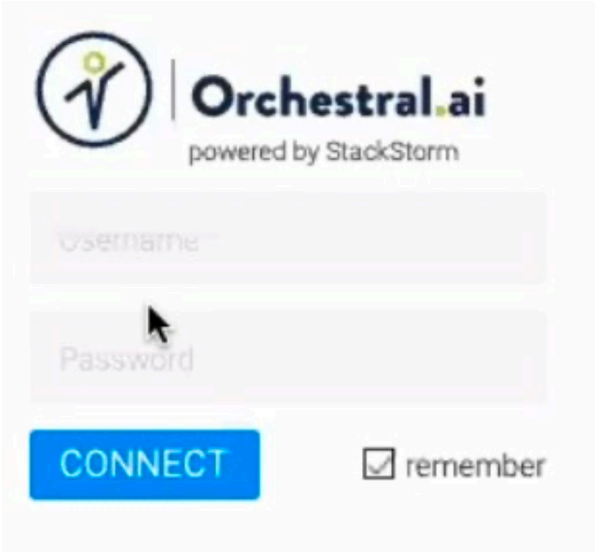


Once the installation has finished, make sure all of the preconfigured packs are installed. Login to Composer using the user and password you created in the install script. Next check the pack list:

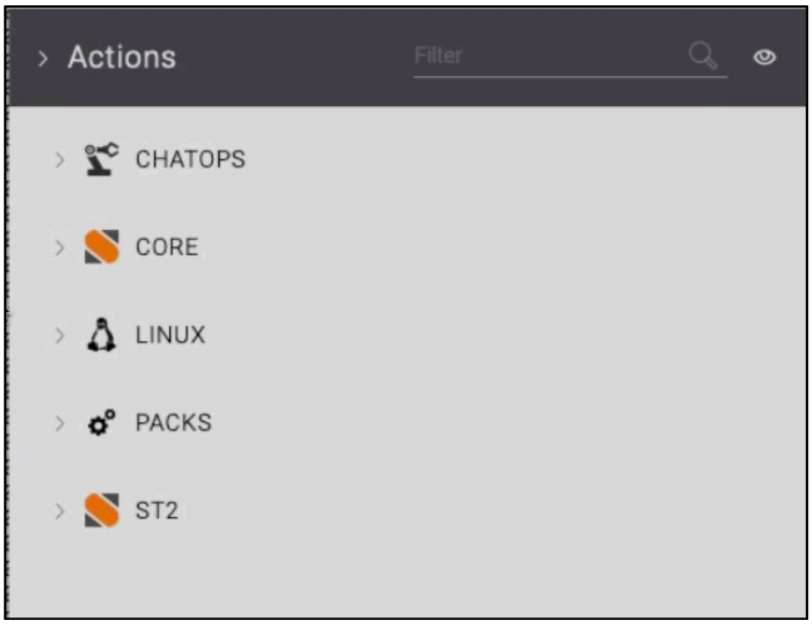
```
st2 login <username>
st2 pack list
```

```
frank@orch-ha-01:~$ st2 pack list
+---------+---------+---------------------------+---------+-------------------+
| ref     | name    | description               | version | author            |
+---------+---------+---------------------------+---------+-------------------+
| chatops | chatops | ChatOps integration pack  | 3.3.0   | StackStorm, Inc.  |
| core    | core    | Basic core actions.       | 3.3.0   | StackStorm, Inc.  |
| default | default | Default pack where all    | 3.3.0   | StackStorm, Inc.  |
|         |         | resources created using   |         |                   |
|         |         | the API with no pack      |         |                   |
|         |         | specified get saved.      |         |                   |
| linux   | linux   | Generic Linux actions     | 3.3.0   | StackStorm, Inc.  |
| packs   | packs   | Pack management           | 3.3.0   | StackStorm, Inc.  |
|         |         | functionality.            |         |                   |
| st2     | st2     | StackStorm utility        | 1.3.1   | StackStorm, Inc.  |
|         |         | actions and aliases       |         |                   |
+---------+---------+---------------------------+---------+-------------------+
```

This can be verified by logging into the Composer UI.
Use ifconfig to obtain the IP address of the Composer server if needed.
Then connect to that address and use the login credentials you entered within the install script to access your Composer instance.



Once you are logged in, you can go to Actions on the top banner which will then show you what packs you currently have installed.



5. Getting to Know the Basics

5.2 Sensors

Sensors are the heart of Composer. These will observe a set of parameters for any change. I.e connected to a Rest API watching to see if a server goes over load. Can poll webhooks, external API's and systems, and then emit a trigger notifying us that a state has changed. Written in python. YAML metadata.

5.3 Triggers

These act as declaration of an Event. Emitted from any form of sensor or webhook. Contain a payload saying, for example, what file changed in a monitored directory or what server sensor went off saying the server is down.

5.4 Rules

IFTTT Recipe. Described in YAML. If this trigger matches a set of criteria, then execute an action. Mapping from a trigger event to what action will be ran. Can be as complex or as basic as needed. Example would be, a rule that says if a server needs more storage then declare that a specific workflow that would add storage to that server be started.

5.5 Actions

Building blocks of automation, single task, any language, metadata files are in YAML, saying here is what Composer is doing, here are the parameters it takes, describing the action, then points to the python script for that action code. Example would be a saved action script that just pings a server.

5.6 Workflows

How we can execute multiple actions in sequence or in parallel. Collection of actions that are utilized together through logic and branching to perform a function. These are also considered actions. So they can be plugged into another workflow as a single step along a larger process. Building block of auto remediation as you will be able to create workflows that can have rules in place to run additional workflows to fix issues as they arise.. YAML file.

5.7 Packs

Collection of actions, workflows, rules, aliases, sensors. They are git repositories. Pack metadata file describes what the pack does, some keywords and who made it. Directory structure, if you put your code in the right spot the pack will handle the rest for you. Installed from git, open source packs are installed from the exchange.stackstorm.org website.

Quick Summary of our event-driven workflow in action:
Sequence flow: Server runs low on storage (event), a sensor that was monitoring that server goes off from the event. That sensor will then emit a trigger, that trigger will then match the current issue against a set of rules. That chosen rule will then declare that an action, more precisely a workflow, should start. That workflow then performs the actions which the rule described the resolve the issue.

6. Contact Us

If you have run into any problems during the installation process, would like a demo for certain use cases or features, or would like to ask Team Orchestral.ai any questions, please reach out to info@orchestral.ai or submit a contact form on https://orchestral.ai. We look forward to hearing from you.