Email Piping

- Configuration
- Set-up the accounts

Configuration

Ticaga retrieves 10 emails every 2 minutes using a non-overlapping scheduling mechanism. This approach ensures that if an IMAP server experiences delayed response times, subsequent scheduled runs are skipped rather than initiating concurrent jobs. This design prevents excessive CPU load and eliminates the risk of duplicate email processing.

Connect via SSH using the username established during Ticaga installation. Our documentation uses `ticaga` or `demo` as example usernames for illustrative purposes. For security best practices, we recommend using a unique username, though the final choice is yours.

```
sudo su ticaga
```

Now we want to set-up the cronjob for Laravel's scheduler:

```
export EDITOR=nano;
crontab -e
```

First we set the editor to nano, this is the basic editor, you can use vi but in our option Nano is better.
Then we edit the crontab.

Paste in the following cronjob:

```
* * * * * cd /var/www/ticaga && php artisan schedule:run >> /dev/null 2>&1
```

Ctrl + X and Y to save and close the editor.

To check that it's set to run, you can run the following command:

```
crontab -l
```

It'll output the above.

Set-up the accounts

This is the important step to pipe into Ticaga, you need to go to:

```
/config/imap.php
```

Find:

```
/*
    |--------------------------------------------------------------------------
    | Available accounts
    |--------------------------------------------------------------------------
    |
    | Please list all IMAP accounts which you are planning to use within the
    | array below.
    |
    */
    'accounts' => [

        'default' => [// account identifier
            'host'  => 'localhost',
            'port'  => 993,
            'protocol'  => 'imap', //might also use imap, [pop3 or nntp (untested)]
            'encryption'    => 'ssl', // Supported: false, 'ssl', 'tls'
            'validate_cert' => true,
            'username' => 'root@example.com',
            'password' => '',
            'authentication' => null,
            'rfc' => 'RFC822', // If you are using iCloud, you might want to set this to
'BODY'
            'proxy' => [
                'socket' => null,
                'request_fulluri' => false,
                'username' => null,
                'password' => null,
            ],
            "timeout" => 30,
```

```
            "extensions" => []
        ],
    ],
```

Please do not modify the default entry, as it is not integrated with Ticaga and serves solely as a reference example. To add your own accounts, insert them immediately above line 31.

```
        'sales' => [
            'host' => 'mail.yourdomain.com',
            'port' => 993,
            'encryption' => 'ssl',
            'validate_cert' => true,
            'username' => 'sales@yourdomain.com',
            'password' => "your_secret_password",
            'authentication' => 'null',
        ],

        'support' => [
            'host' => 'mail.yourdomain.com',
            'port' => 993,
            'encryption' => 'ssl',
            'validate_cert' => true,
            'username' => 'support@yourdomain.com',
            'password' => "your_secret_password",
            'authentication' => 'null',
        ],
```

The username must match the department email address, however you can have more than one email pipe into a department.

So the final result looks like:

```
/*
    |--------------------------------------------------------------------------
    | Available accounts
    |--------------------------------------------------------------------------
    |
    | Please list all IMAP accounts which you are planning to use within the
    | array below.
    |
    */
```

```
    'accounts' => [

        'default' => [// account identifier
            'host'  => 'localhost',
            'port'  => 993,
            'protocol'  => 'imap', //might also use imap, [pop3 or nntp (untested)]
            'encryption'    => 'ssl', // Supported: false, 'ssl', 'tls'
            'validate_cert' => true,
            'username' => 'root@example.com',
            'password' => '',
            'authentication' => null,
            'rfc' => 'RFC822', // If you are using iCloud, you might want to set this to
'BODY'
            'proxy' => [
                'socket' => null,
                'request_fulluri' => false,
                'username' => null,
                'password' => null,
            ],
            "timeout" => 30,
            "extensions" => []
        ],

        'sales' => [
            'host' => 'mail.yourdomain.com',
            'port' => 993,
            'encryption' => 'ssl',
            'validate_cert' => true,
            'username' => 'sales@yourdomain.com',
            'password' => "your_secret_password",
            'authentication' => 'null',
        ],

        'support' => [
            'host' => 'mail.yourdomain.com',
            'port' => 993,
            'encryption' => 'ssl',
            'validate_cert' => true,
            'username' => 'support@yourdomain.com',
            'password' => "your_secret_password",
```

```
            'authentication' => 'null',
        ],
    ],
```

And email fetching will automatically start for the configured accounts, Amazing!