**Guía Rápida de RabbitMQ**

1. **Instalación Erlang**

Para poder instalar RabbitMQ en Windows primero debemos instalar el lenguaje de programación Erlang.

Link: <https://www.erlang.org/downloads>

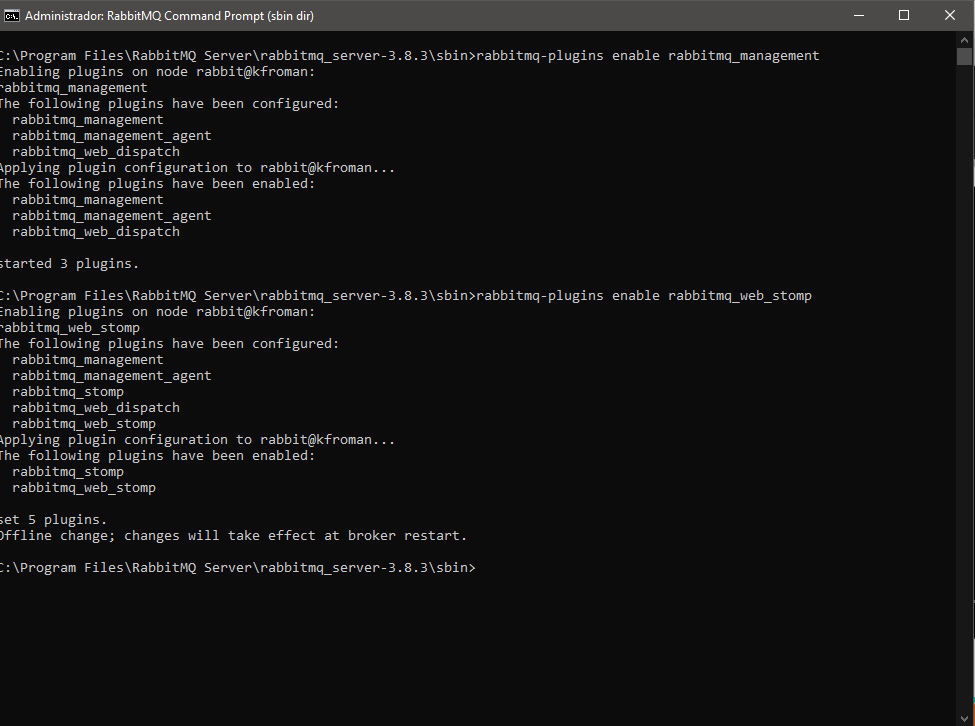
1. **Instalación RabbitMq**

Realizamos la descarga del siguiente enlace:  
Link: <https://www.rabbitmq.com/download.html>

1. **Instalación del plugin Management**Antes de instalar el plugin del management de Rabbit debemos parar el servicio de **RabbitMq Server** para ellos presionamos la tecla Windows y escribimos “**RabbitMQ Service - stop”** y presionamos enter para ejecutarlo. También podemos ubicarlo en la siguiente ruta **“C:\Users\YourUser\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\RabbitMQ Server”**

Una vez que el servicio este parado presionamos la tecla windows y buscamos **“Rabbit command prompt”** presionamos enter y se abrirá la consola de rabbitMq. Seguidamente ejecutamos la siguiente instrucción la cual habilitará el panel de control de rabbitMQ en el navegador.

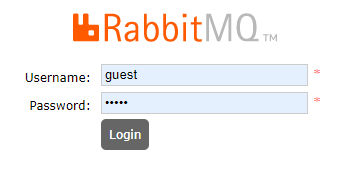
**rabbitmq-plugins enable rabbitmq\_management**



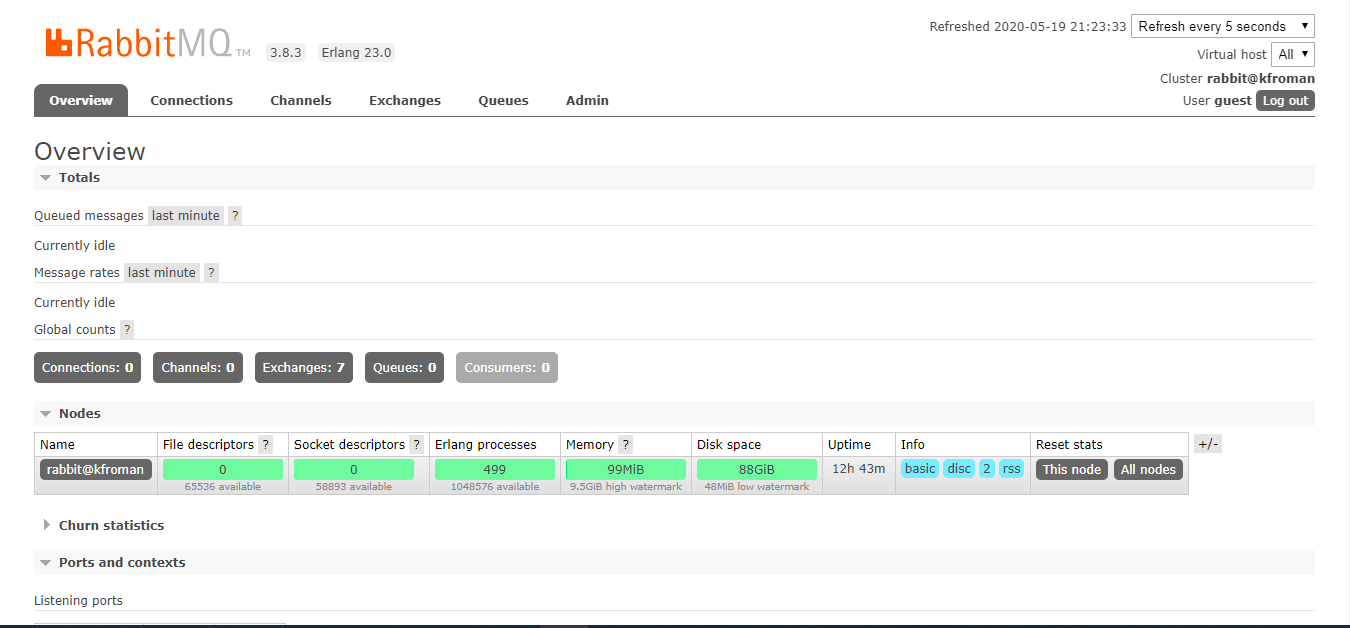
Una vez que la ejecución es exitosa abrimos el navegador y escribimos lo siguiente: <http://localhost:15672> e ingresamos la siguiente información por defecto.

**Username:** guest

**Password:** guest



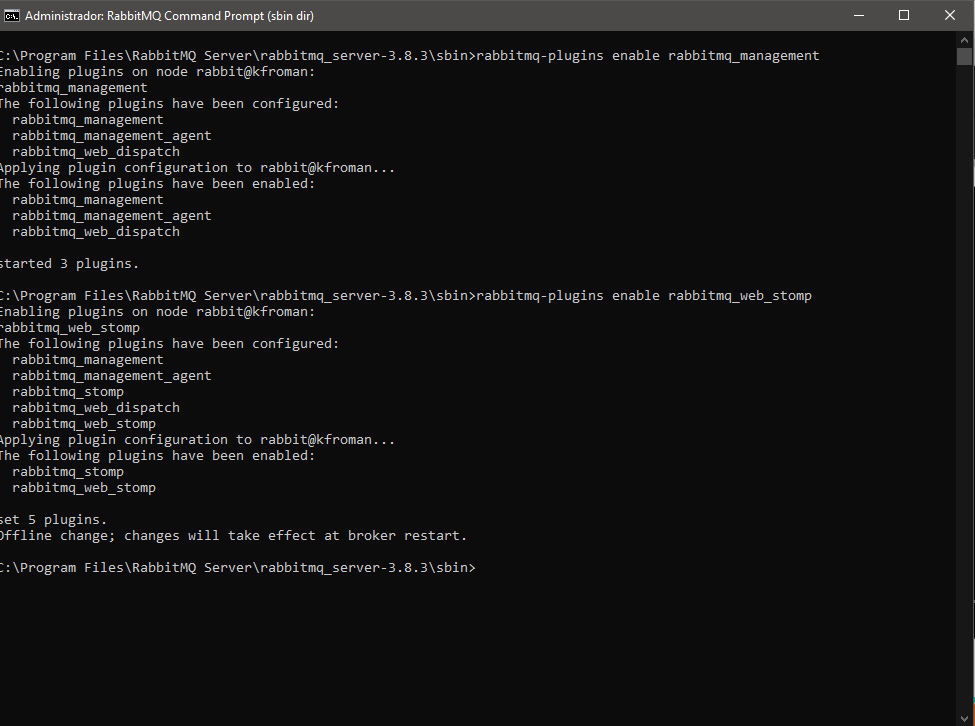
Si el inicio de sesión es exitoso veremos el siguiente panel de control de rabbit MQ

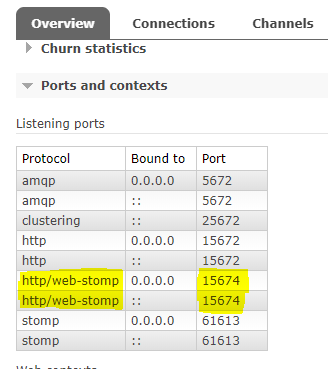


1. **Instalación del plugin Web STOMP**

Abrimos el prompt de comando de rabbit y ejecutamos la siguiente instrucción

**rabbitmq-plugins enable rabbitmq\_stomp**

  
Ingresamos al panel de control de RabbitMq en la pestaña “Overview” verificamos la sección “Listening ports” para comprobar la instalación del plugin



**Ejemplos**

1. **Implementación AMQP**

* Ejemplo en **NodeJS** de cómo enviar un mensaje a **RabbitMQ** mediante el protocolo **AMQP**

// npm install amqplib

// npm express

var express = require('express');

var app = express();

var amqp = require('amqplib/callback\_api');

amqp.connect('amqp://localhost', (err, conn) => {

    conn.createChannel((ee, ch) => {

        var queue = 'FirstQueue';

        var message = { type : '2', content: 'HelloRabbitMq'};

        ch.assertQueue(queue, {durable: false});

        ch.sendToQueue(queue, Buffer.from(JSON.stringify(message)));

        console.log('Message was sent');

    })

    setTimeout(() => {

       conn.close();

       process.exit(0);

    }, 500);

});

app.listen(3001, function () {

  console.log('Example app listening on port 3000!');

});

* Ejemplo en **NodeJS** de cómo leer el mensaje enviado a **RabbitMQ** mediante el ´protocolo **AMQP**

// npm install amqplib

// npm express

var express = require('express');

var app = express();

var amqp = require('amqplib/callback\_api');

amqp.connect('amqp://localhost', (err, conn) => {

    conn.createChannel((ee, ch) => {

        var queue = 'FirstQueue';

        ch.assertQueue(queue, {durable: false});

        console.log(`Waiting for message in ${queue}`);

        ch.consume(queue, (message) => {

            console.log(`Receive ${message.content}`);

        }, {noAck: true} );

    })

});

app.listen(3000, function () {

    console.log('Example app listening on port 3000!');

  });

1. **Implementación Stomp**

* Ejemplo en **NodeJs** de cómo publicar y suscribirse un mensaje en **RabbitMQ** mediante el protocolo **Stomp**

// npm install stomp-client

var Stomp = require("stomp-client");

var queue = "/queue/stompQueue";

var stompClient = new Stomp("127.0.0.1", 61613, "guest", "guest");

client.connect(function(sessionId) {

    client.subscribe(destination, function(body, headers) {

      console.log('This is the body of a message on the subscribed queue:', body);

    });

    client.publish(queue , 'Oh herrow');

});

stompClient.connect(function () {

  stompClient.subscribe(queue , function (data, headers) {

    console.log("GOT A MESSAGE", data, headers);

  });

  setTimeout(function () {

    stompClient.publish(queue , "oh herrow!");

  }, 1000);

  setTimeout(function () {

    stompClient.publish(queue , "wonely...");

  }, 2000);

  setTimeout(function(){

    stompClient.publish(queue , 'so wonely...');

  }, 3000);

  setTimeout(function(){

    stompClient.publish(queue , 'so wonely, so wonely and bwue!');

  }, 4000);

  setTimeout(function () {

    stompClient.disconnect(function () {

      console.log("DISCONNECTED");

    });

  }, 5000);

});

**Bibliografía**

**Fuente** **Amqp**: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/advanced-message-queuing-protocol-amqp/>

**Fuente Stomp**: <https://www.rabbitmq.com/stomp.html>

**Fuente RabbitMq:** <https://www.rabbitmq.com/#features>