Cartostroph! - Server Guide

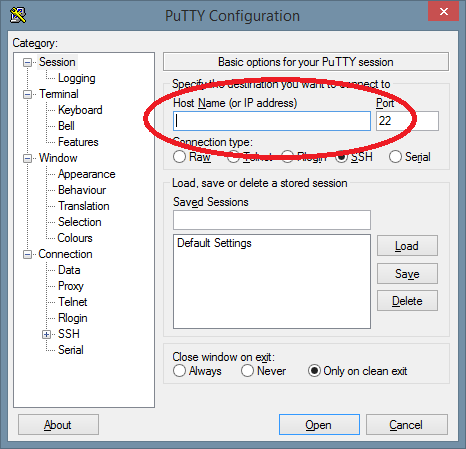
Cartostroph! benötigt einen Server der die Webseite hostet und allen Nutzern bereitstellt. Wir empfehlen dabei einen Linux Server mit einer Version höher als 12.04. Diesen sollten sie sich bei einem bekannten Provider anmieten oder kaufen. Andere Server-Betriebssysteme sind auch kein Problem, jedoch bieten wir keinen direkten Support für diese an und diese Anleitung wird sich direkt auf Linux-Server beziehen.

Als erstes benötigen sie direkten SSH-Zugriff auf ihren Server, dies ist bei gemieteten Servern nicht immer der Fall, deshalb müssen sie konkret bei der Auswahl darauf achten und bestenfalls den Provider nochmal darauf ansprechen.

Wenn sie nun die Zugangsdaten für ihren Server haben, benötigen sie noch ein kleines Tool zur Verbindung.

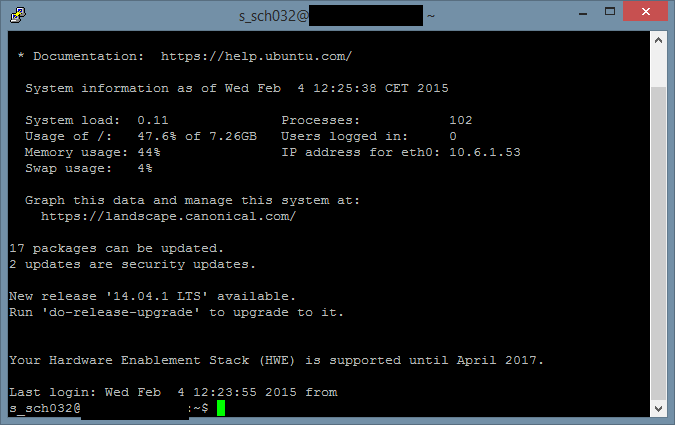
PuTTY download: <http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe>

Führen sie nun einen Doppelklick auf der PuTTY.exe aus. Nun tragen sie die IP-Addresse ihres Servers im rot markierten Feld ein. Mit Port 22 funktionieren die meisten Verbindungen, wenn nicht anders von ihrem Server-Provider angegeben, lassen sie dieses Feld so wie es ist.



Jetzt klicken sie auf ‘Open’ ganz unten in dem Fenster. Nun öffnet sich diese Ansicht.

  
Sie geben jetzt ihren Loginnamen ein, danach kommt eine Abfrage für ihr Passwort. Geben sie dies ein, lassen sie sich nicht verwundern dass keine Zeichen oder Sternchen angezeigt werden. Dies ist normal bei Linux. Nun gelangen sie zu einem Bildschirm der etwa so aussieht.



Jetzt als erstes der Befehl

sudo apt-get update

Ausgeführt werden um die Daten aktuell zu halten.

Als nächstes installieren wir unsere Basis, den Apache Webserver. Das geht ganz einfach und unkompliziert über den Befehl:

sudo apt-get install apache2

Jetzt muss wieder ihr Server-Passwort eingegeben werden und sie müssen die Installation mit J bestätigen wenn danach gefragt wird. Die Installation sollte relativ zügig durchgeführt werden.

Bei diesen Befehlen das gleiche Spiel, Befehl eingeben und bestätigen.

sudo apt-get install apache2-utils

sudo apt-get install postgresql postgresql-contrib phppgadmin

Nun muss die Config Datei von phppgadmin bearbeitet werden. Dies passiert mit dem Befehl:

sudo nano /etc/apache2/conf.d/phppgadmin

Jetzt löschen sie das Zeichen ‘#‘ vor der Zeile „allow from all“ Drücken sie nun „STRG+X“ dies führt zum Speichern der Datei. Bestätigen sie die Abfrage mit J und die zweite mit der Eingabetaste.

Damit nun nicht jeder auf die Seite zugreifen kann, setzen wir noch die Zugriffsrechte für phpPgAdmin. Nutzen sie dafür den Befehl:

sudo nano /etc/apache2/sites-enabled/000-default

Kopieren sie an das Ende des Dokumentes diese Zeilen.

<Directory "/usr/share/phpPgAdmin">

AuthUserFile /etc/phpPgAdmin/.htpasswd

AuthName "Restricted Area"

AuthType Basic

require valid-user

</Directory>

Nachdem sie die Zeilen angefügt haben, Drücken sie nun wieder „STRG+X“ zum Speichern der Datei. Bestätigen sie die Abfrage mit J und die zweite mit der Eingabetaste.

# Datenbank – Nutzer und Datenbank Erstellung

Geben sie nun, in ihrem mit dem Server verbundenen, PuTTY den Befehl

sudo su – postgres

ein. Danach:

createuser –P

Enter name of role to add: dbCarto

Enter password for Cartostroph: \*HIER ihr Passwort\*

Enter it again: \*HIER ihr Passwort\*

Shall the new role be a superuser? (y/n) y

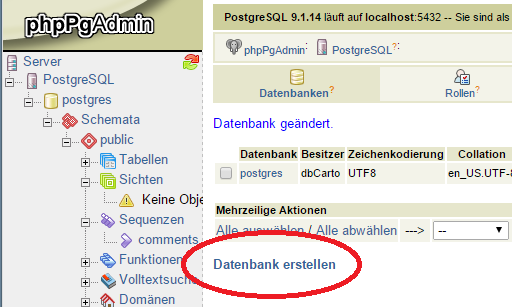
Password:  *‘Geben sie hier das Passwort an womit sie sich in die Datenbank einloggen möchten’*

Jetzt ist ihr Nutzer erstellt und sie können sich auf ihrer PhpPgAdmin Seite einloggen. Dazu gehen sie auf diese Addresse:

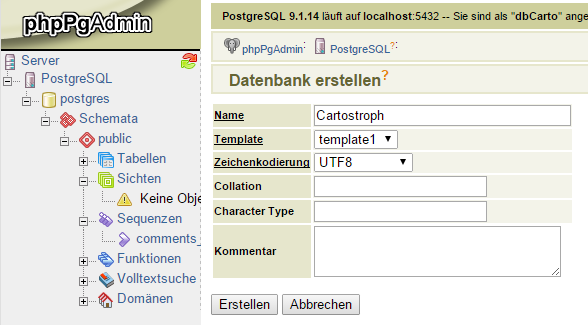
„HIER IHRE SERVERIP“/phppgadmin

Das sollte dann so im Browser aussehen:

Jetzt klicken sie am linken Rand auf „PostgreSQL“ nun öffnet sich eine Abfrage mit Benutzername und Passwort, hier geben sie ihre vorher definierten Nutzerdaten ein. Nun erscheint diese Ansicht, klicken sie nun auf Datenbank erstellen.



Füllen sie nun die Felder genauso aus wie hier gezeigt.



Jetzt öffnet sich wieder die Ansicht von den Datenbanken, klicken sie auf „Cartostroph“ und wählen sie den Reiter „SQL“ aus, direkt neben Schemata. Kopieren sie nun diesen SQL-Befehl

--

-- PostgreSQL database dump

--

SET statement\_timeout = 0;

SET client\_encoding = 'UTF8';

SET standard\_conforming\_strings = on;

SET check\_function\_bodies = false;

SET client\_min\_messages = warning;

--

-- Name: plpgsql; Type: EXTENSION; Schema: -; Owner:

--

CREATE EXTENSION IF NOT EXISTS plpgsql WITH SCHEMA pg\_catalog;

--

-- Name: EXTENSION plpgsql; Type: COMMENT; Schema: -; Owner:

--

COMMENT ON EXTENSION plpgsql IS 'PL/pgSQL procedural language';

SET search\_path = public, pg\_catalog;

SET default\_tablespace = '';

SET default\_with\_oids = false;

--

-- Name: comments; Type: TABLE; Schema: public; Owner: dbCarto; Tablespace:

--

CREATE TABLE comments (

id integer NOT NULL,

name character varying(128) DEFAULT ''::character varying NOT NULL,

email character varying(255) DEFAULT ''::character varying NOT NULL,

body text NOT NULL,

dt timestamp without time zone DEFAULT now() NOT NULL,

rating integer,

page\_id text,

leftpoint point,

rightpoint point

);

ALTER TABLE public.comments OWNER TO "dbCarto";

--

-- Name: comments\_id\_seq; Type: SEQUENCE; Schema: public; Owner: dbCarto

--

CREATE SEQUENCE comments\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public.comments\_id\_seq OWNER TO "dbCarto";

--

-- Name: comments\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: dbCarto

--

ALTER SEQUENCE comments\_id\_seq OWNED BY comments.id;

--

-- Name: nutzer; Type: TABLE; Schema: public; Owner: dbCarto; Tablespace:

--

CREATE TABLE nutzer (

name character varying(30) NOT NULL,

passwort character varying(1000) NOT NULL,

ort character varying(30),

plz character varying(25),

land character varying(25)

);

ALTER TABLE public.nutzer OWNER TO "dbCarto";

--

-- Name: topic; Type: TABLE; Schema: public; Owner: dbCarto; Tablespace:

--

CREATE TABLE topic (

url\_top character varying(1000) NOT NULL,

text character varying(1000) NOT NULL,

bewertung integer,

hyperlink character varying(1000),

anfangsdatum date,

enddatum date,

kategorie character varying(20),

titel character varying(50),

"position" point,

autor character varying(30) NOT NULL,

tag character varying(500),

lpoint point,

rpoint point

);

ALTER TABLE public.topic OWNER TO "dbCarto";

--

-- Name: id; Type: DEFAULT; Schema: public; Owner: dbCarto

--

ALTER TABLE ONLY comments ALTER COLUMN id SET DEFAULT nextval('comments\_id\_seq'::regclass);

--

-- Data for Name: comments; Type: TABLE DATA; Schema: public; Owner: dbCarto

--

COPY comments (id, name, email, body, dt, rating, page\_id, leftpoint, rightpoint) FROM stdin;

\.

--

-- Name: comments\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: dbCarto

--

SELECT pg\_catalog.setval('comments\_id\_seq', 1, false);

--

-- Data for Name: nutzer; Type: TABLE DATA; Schema: public; Owner: dbCarto

--

COPY nutzer (name, passwort, ort, plz, land) FROM stdin;

Gast 123

Guest 123

Gość 123

\.

--

-- Data for Name: topic; Type: TABLE DATA; Schema: public; Owner: dbCarto

--

COPY topic (url\_top, text, bewertung, hyperlink, anfangsdatum, enddatum, kategorie, titel, "position", autor, tag, lpoint, rpoint) FROM stdin;

\.

--

-- Name: comments\_pkey; Type: CONSTRAINT; Schema: public; Owner: dbCarto; Tablespace:

--

ALTER TABLE ONLY comments

ADD CONSTRAINT comments\_pkey PRIMARY KEY (id);

--

-- Name: nutzer\_pkey; Type: CONSTRAINT; Schema: public; Owner: dbCarto; Tablespace:

--

ALTER TABLE ONLY nutzer

ADD CONSTRAINT nutzer\_pkey PRIMARY KEY (name);

--

-- Name: topic\_pkey; Type: CONSTRAINT; Schema: public; Owner: dbCarto; Tablespace:

--

ALTER TABLE ONLY topic

ADD CONSTRAINT topic\_pkey PRIMARY KEY (url\_top);

--

-- Name: topic\_autor\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: dbCarto

--

ALTER TABLE ONLY topic

ADD CONSTRAINT topic\_autor\_fkey FOREIGN KEY (autor) REFERENCES nutzer(name) ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: public; Type: ACL; Schema: -; Owner: postgres

--

REVOKE ALL ON SCHEMA public FROM PUBLIC;

REVOKE ALL ON SCHEMA public FROM postgres;

GRANT ALL ON SCHEMA public TO postgres;

GRANT ALL ON SCHEMA public TO PUBLIC;

--

-- PostgreSQL database dump complete

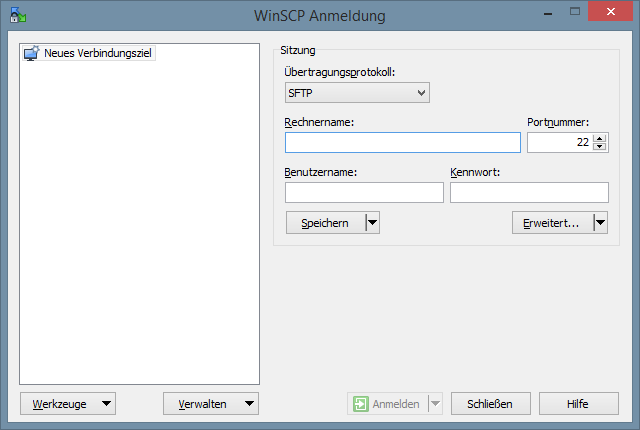
--

# Webseite hochladen und Config.php erstellen

Für das Hochladen der Webseite empfiehlt sich ein Programm namens WinSCP (für Windows) <http://winscp.net/download/winscp556setup.exe> oder Fugu (MacOS) <http://sourceforge.net/projects/fugussh/files/Release/fugu-1.2.0/>

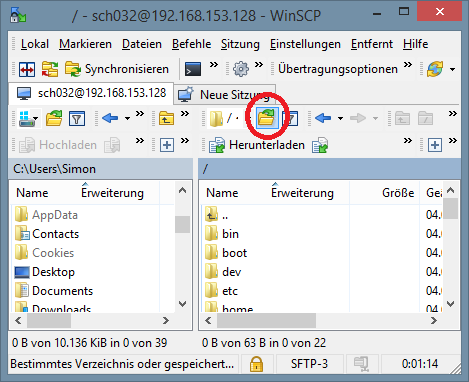
Ich zeige ihnen wie es mit WinSCP geht mit Fugu ist es quasi identisch.

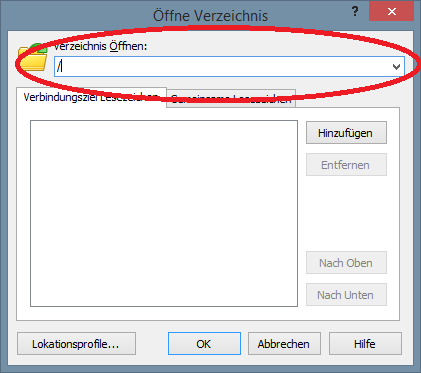
Wenn die Installation abgeschlossen ist, starten sie WinSCP. Es erscheint ein Fenster das so aussieht:



Tragen sie ihre Server IP-Addresse wie bei PuTTY unter Rechnername ein. Als Benutzername und Kennwort werden die Serverseitigen Daten benötigt, nicht die von der Datenbank! Diese Daten haben sie auch in PuTTY verwendet.

Jetzt öffnet sich die Verbindung zum Server und es wird ungefähr so aussehen. Klicken sie auf den Button der rot umkreist ist.

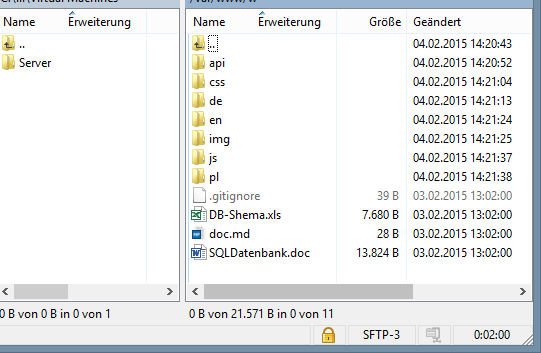


Jetzt öffnet sich dieses Fenster: 

Geben „/var/www/“ in die umkreiste Zeile ein. Bestätigen sie mit einem Klick auf ‚OK‘.

Nun sind sie im Verzeichnis der Webseite. Es sollte dort nur ein Ordner exisiteren, dieser trägt die Beschriftung ‚html‘. Er kann gelöscht werden, da er im weiteren Verlauf nicht mehr benötigt wird.

Jetzt werden noch die Daten der Webseite benötigt, diese sind frei zugänglich auf GitHub. Über diesen Link: <https://github.com/mruhe/cartostroph/releases> kommt man zu den neusten Versionen. Jetzt downloaden sie die neuste Version als .zip-Datei. Extrahieren sie den Ordner und kopieren sie den Inhalt per Drag-and-Drop in den „www“-Ordner den wir zuvor bei WinSCP geöffnet haben. Ihr Ordner sollte nun ungefähr so aussehen. (Hängt von der Version bei Github ab)



Jetzt muss noch der Zugriff auf die Datenbank geregelt werden, erstellen sie dazu eine neue Datei mit dem Namen ‚config.php‘.

Dort kopieren sie diese Zeilen rein und geben das Passwort an der entsprechenden Stelle ein.

<?php

$config["host"] = "localhost";

$config["dbname"] = "Cartostroph";

$config["user"] = "dbCarto";

$config["password"] = "\*\*IHR PASSWORT FÜR DIE DATENBANK\*\*";

$config["connection"] = "host=localhost dbname=Cartostroph user=dbCarto password=\*\*IHR PASSWORT FÜR DIE DATENBANK\*\*";

?>

Jetzt kopieren sie die config.php Datei in den ‚en , pl und den de‘ Ordner. In jedem der drei Ordner existiert auch noch ein Ordner mit dem Namen ‚commentSys‘ dort fügen sie die config.php auch noch ein. Am Ende haben sie sechs Dateien erstellt. Dies erscheint zunächst unnötig kompliziert, da die Dateien alle den selben Inhalt besitzen. Es macht es jedoch nachträglich einfacher wenn man z.B für die englische Version eine andere Datenbank benutzen möchte.

# Webseite aufrufen

Sie können ihre persönliche Version von Cartostroph! nun über ihre Server IP aufrufen.