**DOCKER**

[**https://docs.docker.com/introduction/understanding-docker/**](https://docs.docker.com/introduction/understanding-docker/)

[**https://docs.docker.com/userguide/**](https://docs.docker.com/userguide/)

[**https://docs.docker.com/installation/#installation**](https://docs.docker.com/installation/#installation)

[**https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-getting-started**](https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-getting-started)

[**https://www.youtube.com/watch?v=umJYDAYxZys**](https://www.youtube.com/watch?v=umJYDAYxZys)

**ht8tps://docs.docker.com/articles/networking/**

[**https://docs.docker.com/userguide/dockerizing/**](https://docs.docker.com/userguide/dockerizing/)

[**https://docs.docker.com/userguide/dockerlinks/**](https://docs.docker.com/userguide/dockerlinks/)

**\*https://developer.basespace.illumina.com/docs/content/documentation/native-apps/manage-docker-image**

[**https://docs.docker.com/articles/dockerfile\_best-practices/**](https://docs.docker.com/articles/dockerfile_best-practices/)

[**https://docs.docker.com/terms/image/#base-image**](https://docs.docker.com/terms/image/#base-image)

**\*https://learn.tutum.co/article/2/continuous-integration-using-docker-maven-and-jenkins**

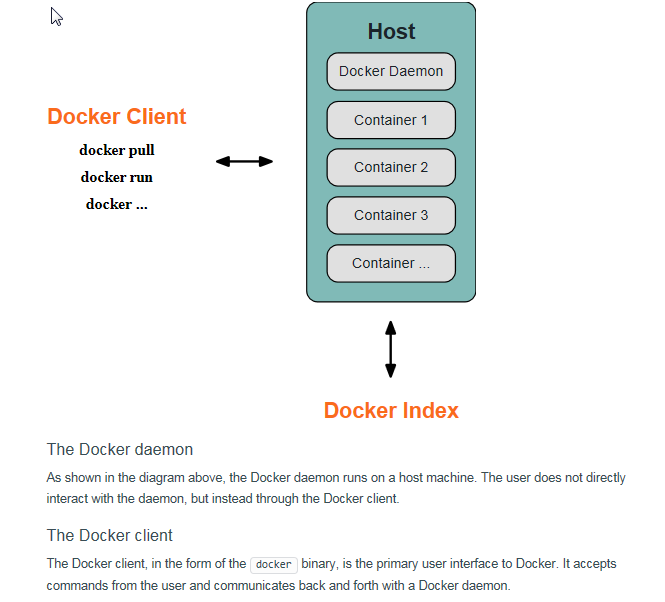
**\*http://container-solutions.com/2015/03/running-docker-in-jenkins-in-docker/**

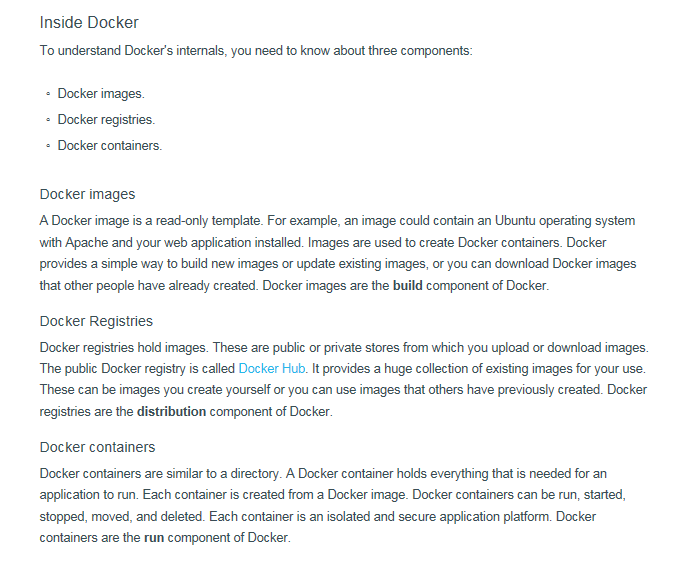
\*\*\*\*http://olavgg.com/post/88786078128/running-a-docker-container-that-starts-a

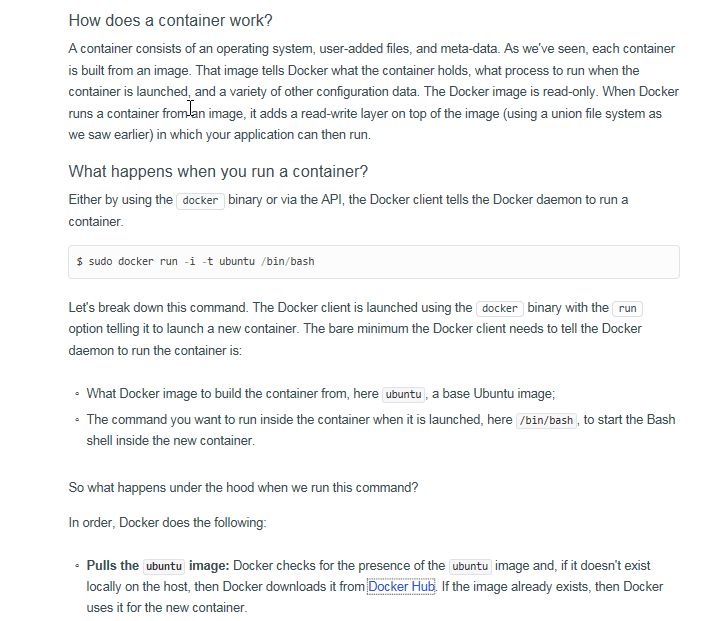
|  |
| --- |
| http://planet.jboss.org/post/9\_docker\_recipes\_for\_java\_ee\_applications\_tech\_tip\_80 |

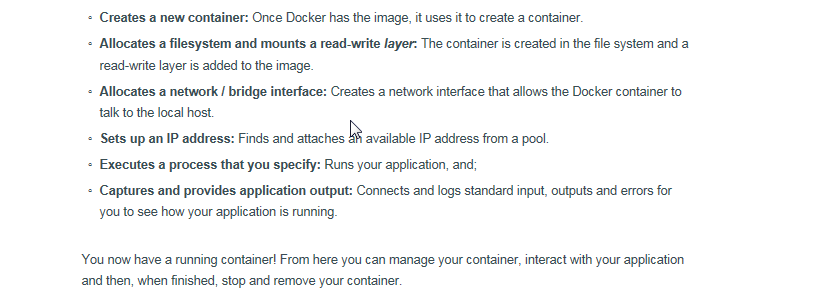
What is Docker's architecture?

Docker uses a client-server architecture. The Docker *client* talks to the Docker *daemon*, which does the heavy lifting of building, running, and distributing your Docker containers. Both the Docker client and the daemon *can* run on the same system, or you can connect a Docker client to a remote Docker daemon. The Docker client and daemon communicate via sockets or through a RESTful API.



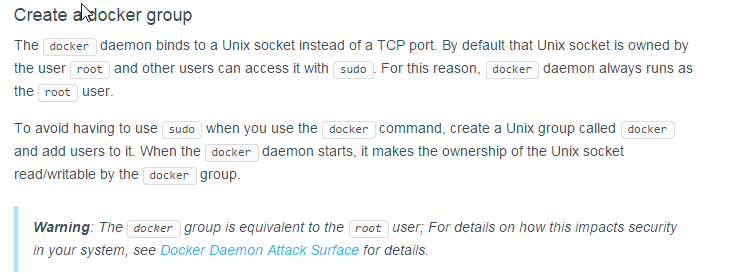


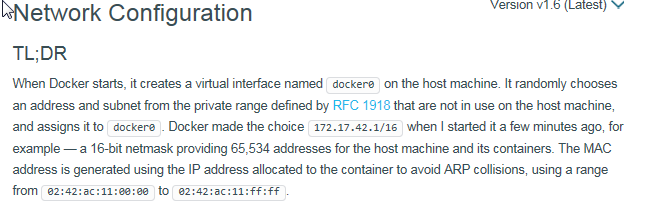


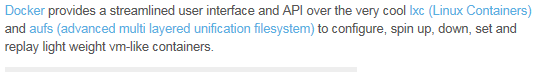


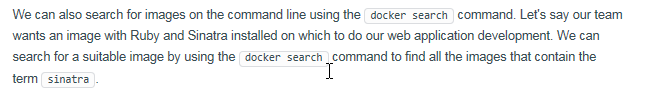
|  |
| --- |
| These instructions are stored in a file called a Dockerfile. Docker reads this Dockerfile when you request a build of an image, executes the instructions, and returns a final image. |
| Linking only works if all the containers are running on the same host |

|  |  |
| --- | --- |
| current kernel version | uname -r |
| To List the Commands in Docker | docker |
| To add user | **adduser dockertest** |
| Lists containers | docker ps |
| list images | docker images |
| List of Container | docker ps -a |
| -s | docker ps -a –s (-a makes sure you see already closed containers ,–s gives you the container file size) |
| To show latest Container | docker ps --latest |
| Shows us the standard output of a container | docker logs |
| To Start and Stop Docker Service | * service docker stop * service docker start |
| Stops running containers | docker stop |
| Docker Version | docker version |
| -d | The -d flag which tells Docker to run the container in the background |
| -P | tells Docker to map any required network ports inside our container to our host. |
| -a | To see stopped containers too. |
| Search an Image | docker search sinatra (Here sinatra is a user image) |
| Pull image | docker pull training/sinatra |
| To Stop a container | docker stop ab35cf9ba9f0 (Here ab35cf9ba9f0 container id if sinatra) |
| To Start a container | docker start ab35cf9ba9f0 |
| -m | The -m flag allows us to specify a commit message, much like you would with a commit on a version control system |
| -a | The -a flag allows us to specify an author for our update. |
| To get details of a container | docker inspect aa91927c81ab (Here aa91927c81ab docker ID) |
| To get IP address of Container | docker inspect -f '{{ .NetworkSettings.IPAddress }}' aa91927c81ab(Here aa91927c81ab docker ID) |
| To enter running container | docker exec -it caf484e836e5 bash(Here caf484e836e5 is container ID)  or  docker top  example :docker exec -it test ps aux  https://docs.docker.com/reference/builder/ |
| To find the portbinding | docker port caf484e836e5(Here caf484e836e5 is container ID)  or  docker port caf484e836e5 5000 (5000 is the port number assigned by the container) |
| To check the logs | docker logs -f caf484e836e5 |
| To check the process inside the container | docker top caf484e836e5 |
| To remove the container | docker rm caf484e836e5 or nostalgic\_morse  (Here nostalgic\_morse is Docker name) |
| You can name your container by using the --name flag | docker run -d -P --name web training/webapp python app.py |
| To find Name of a container | docker inspect -f "{{ .Name }}" aed84ee21bde |
| Linking two contanier | --link <name or id>:alias    or  docker run -d -P --name web --link db:db training/webapp python app.py  (Here db is name of one container and web of another,wb container is  created and linked to db container ) |
| Inspect your linked containers | docker inspect -f "{{ .HostConfig.Links }}" web  [/db:/web/db] |
| The env command to list the specified container's environment variables | docker run --rm --name web2 --link db:db training/webapp env |
| CMD | The CMD is the command that will be executed when a container is started |
| To remove Docker a Image | docker rmi |
| The docker tag command takes the ID of the image, here 5db5f8471261, and our user name, the repository name and the new tag | docker tag 5db5f8471261 ouruser/sinatra:devel |
| To Check Port number | netstat -ntlup|grep 4243(Here 4243 is the port number) |
|  | iptables-save |
| --rm | removes the container as soon as you stop it |
| -p | • forwards the port to your host (or if you use boot2docker to this IP) |
| -it | allows interactive mode, so you see if something get's deployed |

. 







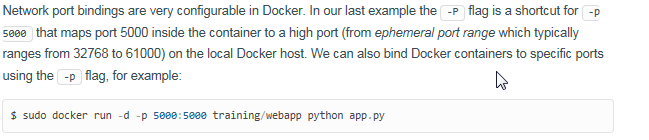
**Two Type of Images**



**To Commit Docker**

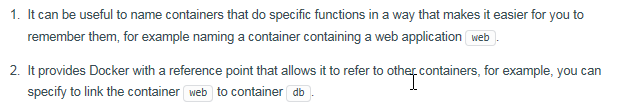


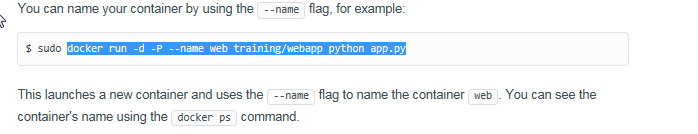
**To PORT BINDING**

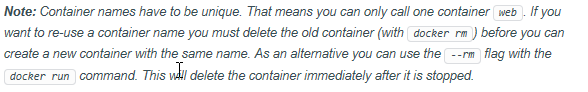


**Naming a Container**

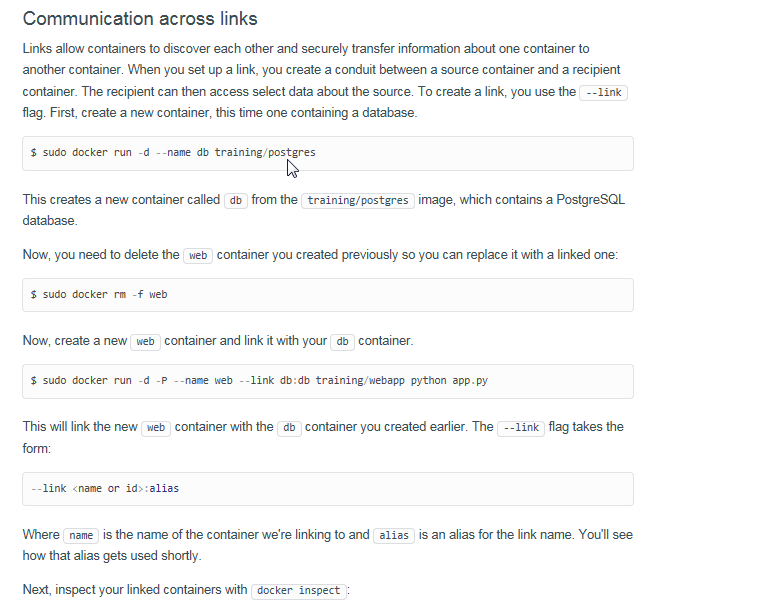
This naming provides two useful functions:

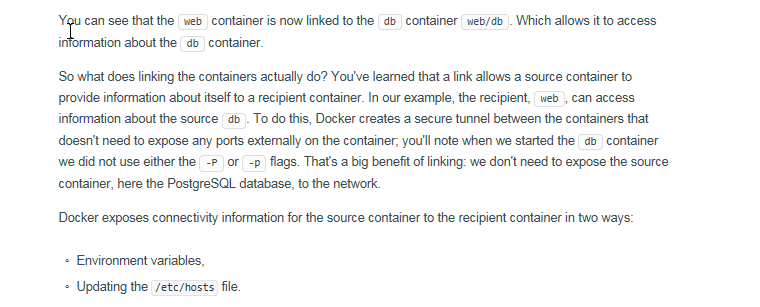


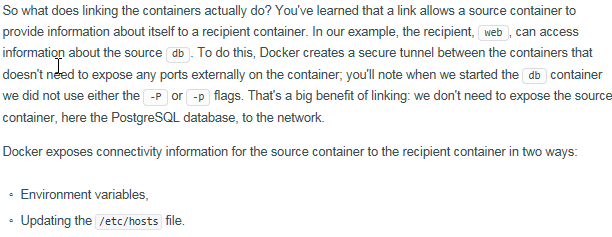


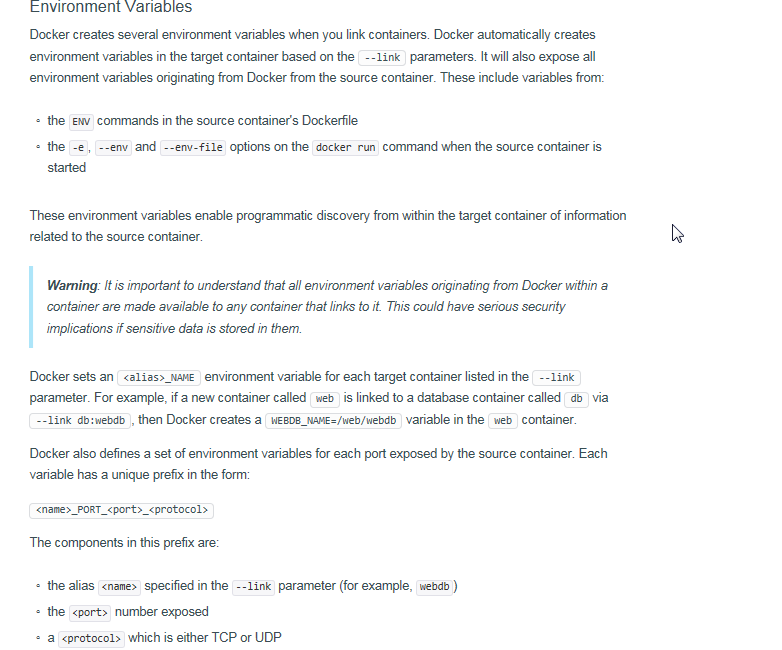


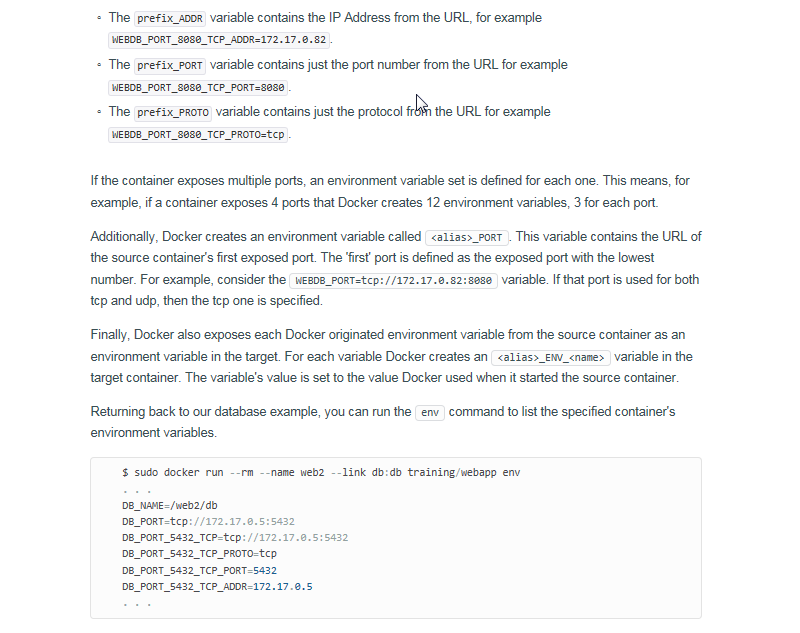
**Linking two Container**

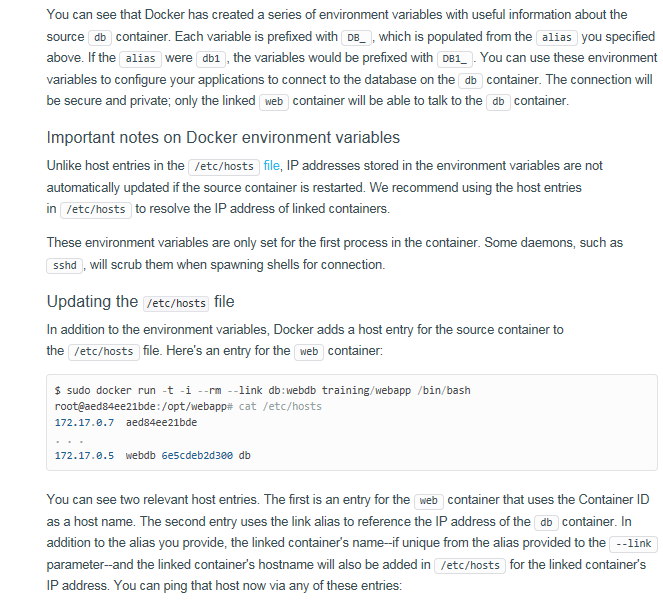


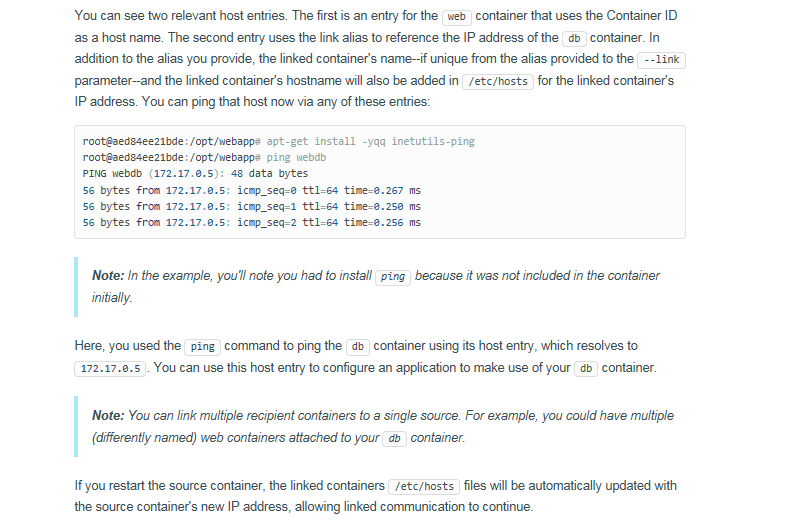


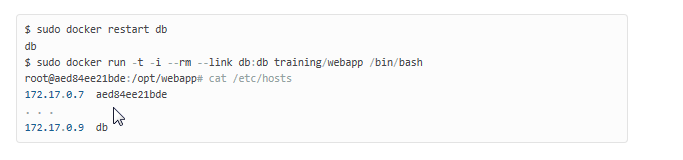




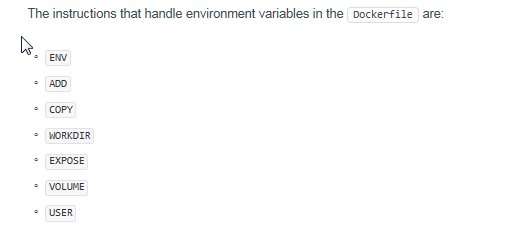


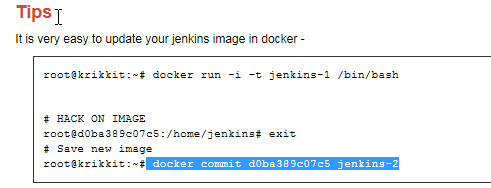






**DockerFile Commands**





**Docker will never make changes to your system iptables rules if you set --iptables=false when the daemon starts. Otherwise the Docker server will add a default rule to the FORWARD chain with a blanket ACCEPT policy if you retain the default --icc=true, or else will set the policy to DROP if --icc=false.**

**It is a strategic question whether to leave --icc=true or change it to --icc=false (on Ubuntu, by editing the DOCKER\_OPTS variable in /etc/default/docker and restarting the Docker server) so that iptables will protect other containers — and the main host — from having arbitrary ports probed or accessed by a container that gets compromised.**

**Dockerfile\_Tomcat**

1.export http\_proxy=http://10.53.130.192:80

2.export https\_proxy=https://10.53.130.192:80

3.apt-get update

4. apt-get install wget –y

5. unset http\_proxy

6.wget <http://10.53.77.137:6060/html/repos/jdk-7u79-linux-x64.tar.gz>

7. mkdir /usr/java

8.mv jdk-7u79-linux-x64.tar.gz /usr/java/

9. tar -xvzf /usr/java/jdk-7u79-linux-x64.tar.gz -C /usr/java

10. update-alternatives --install /usr/bin/java java /usr/java/jdk1.7.0\_79/jre/bin/java 200000

11. update-alternatives --install /usr/bin/javaws javaws /usr/java/jdk1.7.0\_79/jre/bin/javaws 200000

12.update-alternatives --install /usr/lib64/mozilla/plugins/libjavaplugin.so libjavaplugin.so.x86\_64 /usr/java/jdk1.7.0\_79/jre/lib/amd64/libnpjp2.so 200000

 13.update-alternatives --install /usr/bin/javac javac /usr/java/jdk1.7.0\_79/bin/javac 200000

14.update-alternatives --install /usr/bin/jar jar /usr/java/jdk1.7.0\_79/bin/jar 200000

15.wget <http://10.53.77.137:6060/html/repos/apache-tomcat-7.0.55.tar.gz>

16.mv apache-tomcat-7.0.55.tar.gz /opt

17.tar -xvzf /opt/apache-tomcat-7.0.55.tar.gz -C /opt

18.CMD cd /opt/apache-tomcat-7.0.55/bin && sh startup.sh

**FIG**

**Docker orchestration using Fig**

Installation PathURL: <http://www.fig.sh/install.html>