Resource: input to the jobs

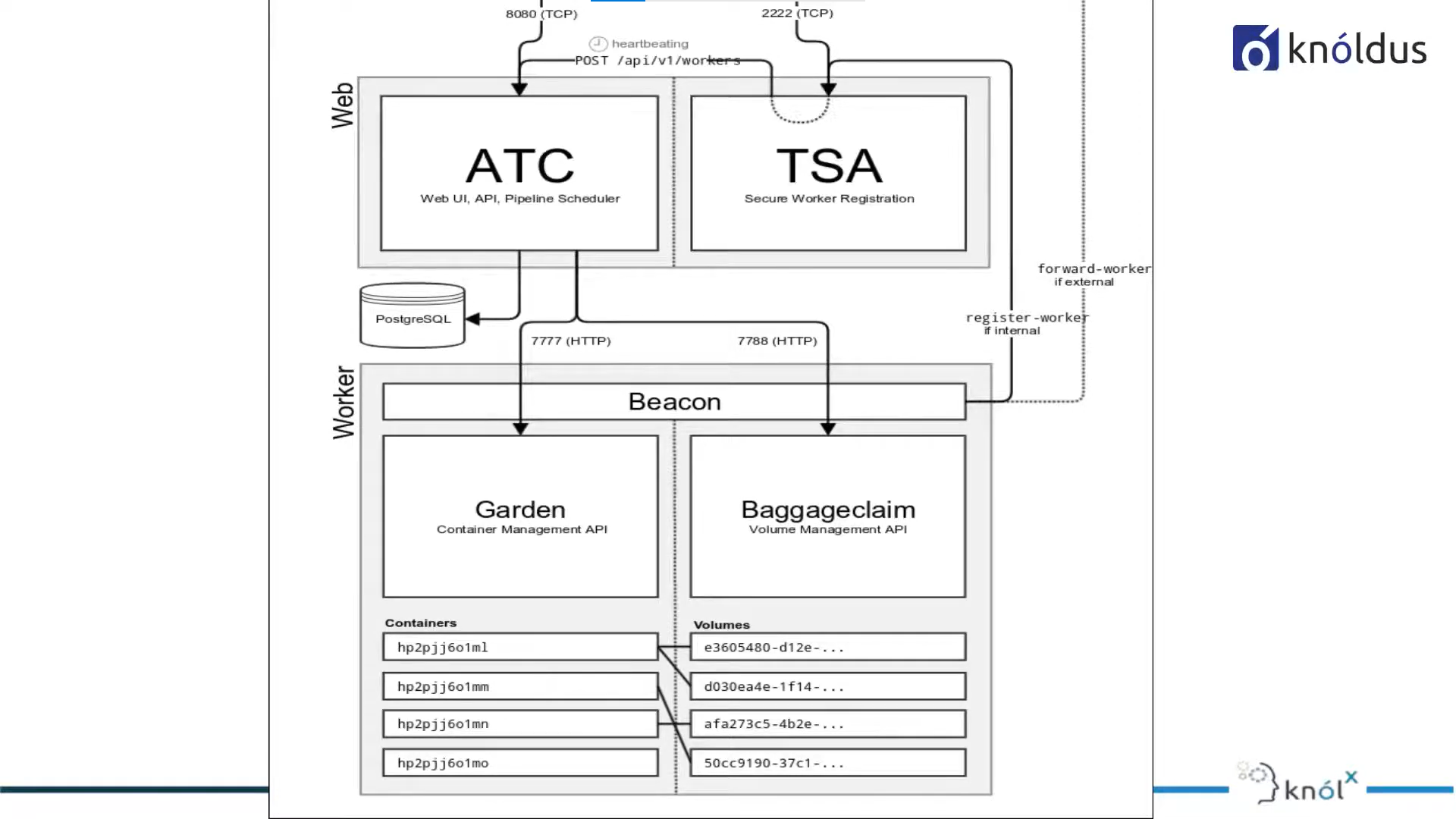
Task: some instructions (smallest unit execution)

Iobs: build plan (set of tasks)

Version is state of resource

Vars:

Syntax: ((source-name:secret-path.secret-field))



A screenshot of a computer

Description automatically generated with low confidence

**Integrate the fly cli and the conconcourse web:**

fly -t example login -c http://localhost:8080/ -u test -p test

**set the pipeline:**

fly -t example set-pipeline -p example-pipeline -c <file-path>

Resourse:

It is just source, it won’t pull RS

* Name: <identifier>

Type: < CC will decide what are the operations to be performed like clone, pull based on type>

Source: <source of the type>

jobs:

- name: <identifier>

plan: < sequence of tasks or steps>

- get: <name of resource> CC will fetch the latest version of the RS, save it as artifact and made it available to the all the task under the specified job. Then the tasks can use it using input field.

passed: [previous-job-name]<executes after the job executes

- set\_pipeline: pipeline's name which you want to trigger

file: pipeline's config path

- task: <identifier>

Config: < contains all field in a task config>

Platform: <which worker node to select based on os>

inputs:

- name: <name of artifacts available from the get or config.output from the

previous task>

image\_resource:<where the present task should run>

type:

source:

run:<commands to run in the container or in the platform>

path:

file: <path of the yaml file which contains config part (alternate to the config)>

vars: <to pass variable to the external file>

image: get-artifact< if you want to use the same image from the RS, alternate to image\_RS>

put:<to update the resource>

**set\_pipeline:**

file: [RS-artifact name]/file-path <the config file for the new job in the new pipeline>

instance\_vars