You are in: DevOps Technical Workshop Wiki > V2 Lab VI - Return of the Dashboards > V2 Lab 6 - Steps 2C

V2 Lab 6 - Steps 2C

Like | Updated 21 December 2016 by Spring, Donal | Tags: *None*

(C) Traffic light jobs

The objective of this lab is to monitor each layer of our applications health with a simple sniff test. To do this, a simple bash script has been written which sends a simple request to the front end to see if the app is up, the server layer and the mongodb. These will be created for each environment (si and production)

– On Jenkins, create a new job called `tl-mongodb-si`

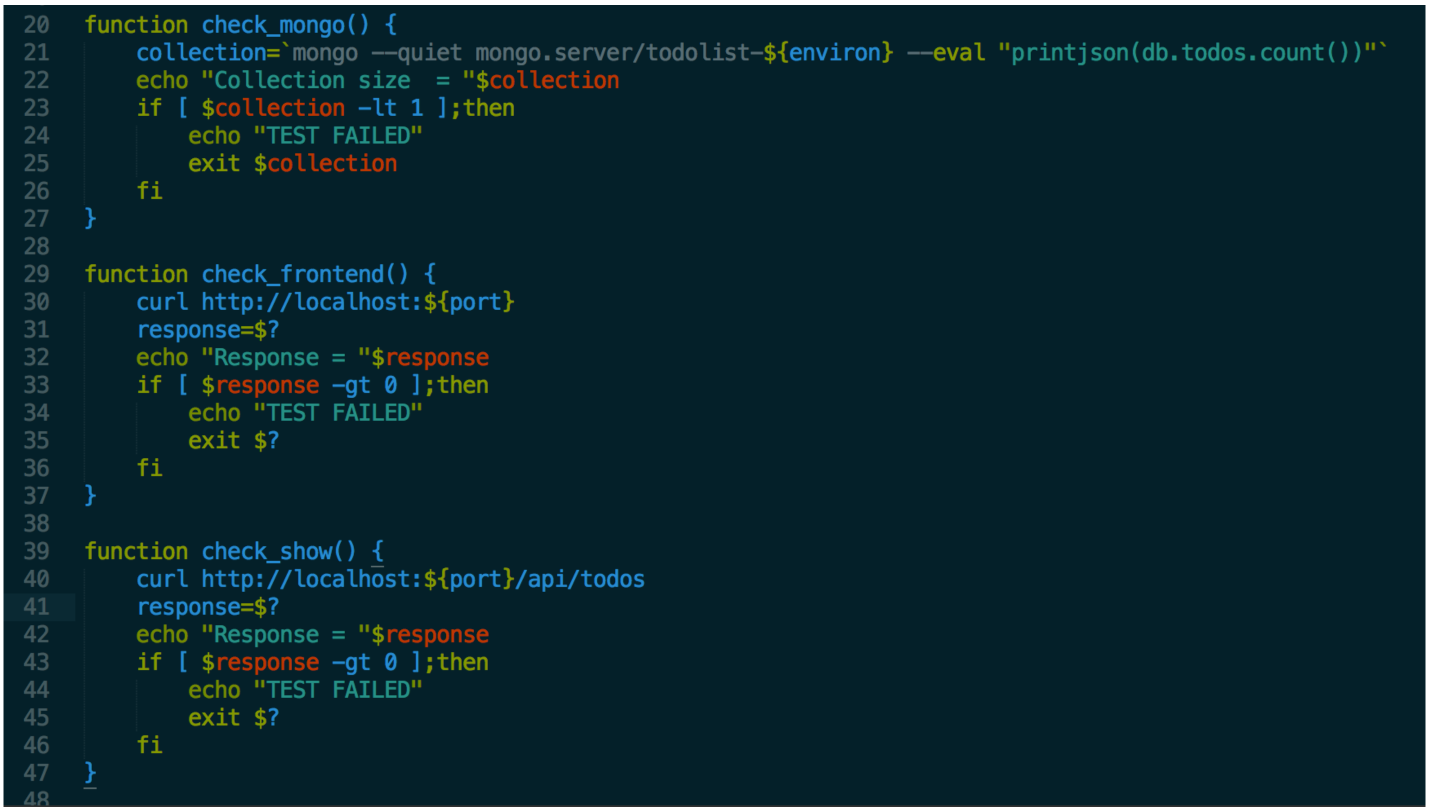
– In the *Source Code Management* tab, set it to the lab-material repository `ssh://git@localhost/home/git/lab-material.git` and set the credentials to the `jenkins (id_rsa)` file.

– Add a *Build Trigger* > *Build Periodically* and set it to run every 5 minutes `H/5 * * * *`

– Add a *Build Step* to *Execute Shell*. Put this in the box to run the script in the lab material repository. The first param of the script is the environment to point to eg si, the second is the function to run in the file eg check_mongo. Save once done.

```
./scripts/traffic-lights.sh si check_mongo
```

The script contains three functions for testing each layer of the stack as shown below



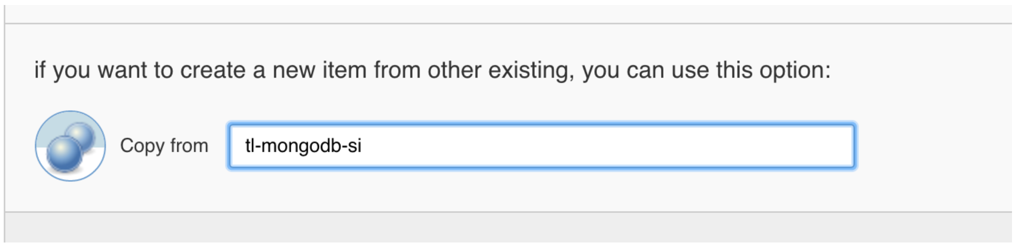
```
function check_mongo() {
    collection=`mongo --quiet mongo.server/todolist-${environ} --eval "printjson(db.todos.count())"`
    echo "Collection size  = "$collection
    if [ $collection -lt 1 ];then
        echo "TEST FAILED"
        exit $collection
    fi
}

function check_frontend() {
    curl http://localhost:${port}
    response=$?
    echo "Response = "$response
    if [ $response -gt 0 ];then
        echo "TEST FAILED"
        exit $?
    fi
}

function check_show() {
    curl http://localhost:${port}/api/todos
    response=$?
    echo "Response = "$response
    if [ $response -gt 0 ];then
        echo "TEST FAILED"
        exit $?
    fi
}
```

– Run the job to validate it is working correct.

– Create two more tl-si jobs calling each of the functions defined above ie `tl-frontend-si` & `tl-api-show-si`. Copy the configuration from the `tl-mongodb-si` job when creating the new ones for ease. In the execute shell step swap the `check_mongo` for either `check_frontend` for the `tl-frontend-si` job AND `check_show` for the `tl-api-show-si`.


if you want to create a new item from other existing, you can use this option:

Copy from tl-mongodb-si

– Do this again for the *-production* environment ie create three more jobs (`tl-mongodb-prod` AND `tl-frontend-prod` AND `tl-api-show-prod`) but configuring the build task to be the following for each of these

tl-mongodb-prod job build step:

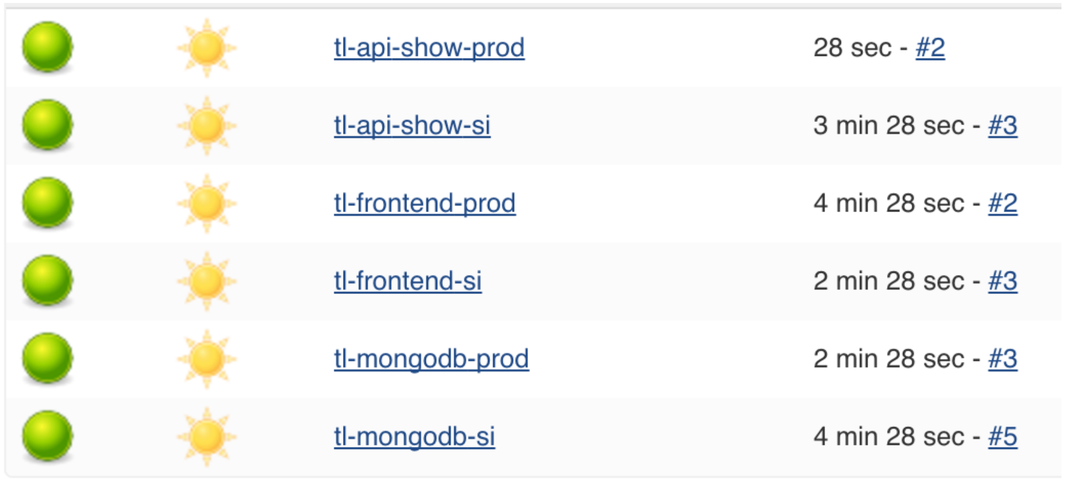
```
./scripts/traffic-lights.sh prod check_mongo
```

tl-frontend-prod job build step:

```
./scripts/traffic-lights.sh prod check_frontend
```

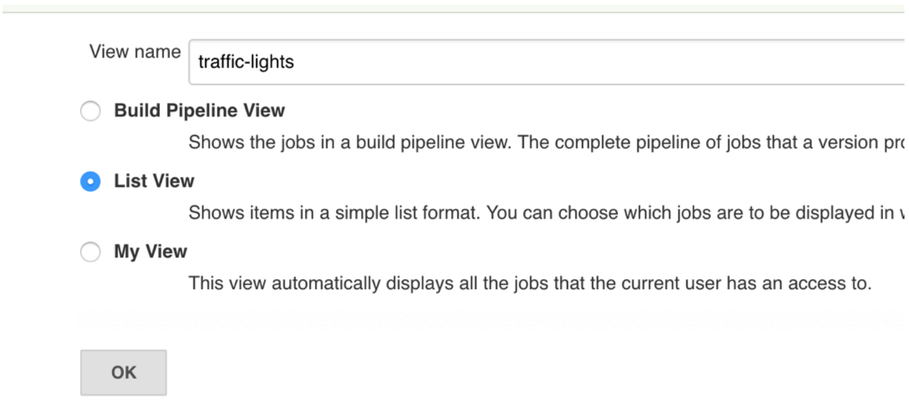
tl-api-show-prod job build step:

```
./scripts/traffic-lights.sh prod check_show
```



●	☀	tl-api-show-prod	28 sec - #2
●	☀	tl-api-show-si	3 min 28 sec - #3
●	☀	tl-frontend-prod	4 min 28 sec - #2
●	☀	tl-frontend-si	2 min 28 sec - #3
●	☀	tl-mongodb-prod	2 min 28 sec - #3
●	☀	tl-mongodb-si	4 min 28 sec - #5

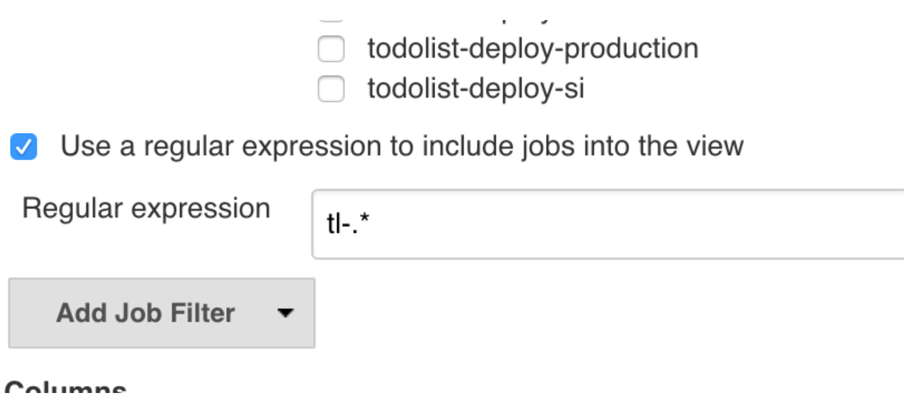
– Create a List View for these jobs to keep them together. Hit the + icon on the top of the list of jenkins jobs. Name the new view something sensible such as `traffic-lights` Use the regex `tl-.*` to gather them all


View name traffic-lights

Build Pipeline View
Shows the jobs in a build pipeline view. The complete pipeline of jobs that a version pro

[✓ List View]
Shows items in a simple list format. You can choose which jobs are to be displayed in v

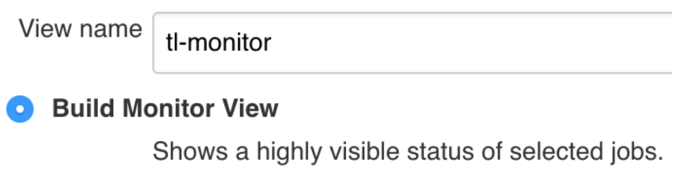
My View
This view automatically displays all the jobs that the current user has an access to.


☐ todolist-deploy-production
☐ todolist-deploy-si

☑ Use a regular expression to include jobs into the view

Regular expression: tl-.*

– Finally, create an information radiator for the jobs, create a new Build Monitor view and add the jobs to it. Hit the + icon again on Jenkins homepage, then create *Build Monitor View* with a sensible name such as `tl-monitor`. Use the regex `tl-.*` to gather them all as before.


View name: tl-monitor

● **Build Monitor View**

Shows a highly visible status of selected jobs.

– The built view should look like this. Experiment with stopping / starting various parts of the stack. Use `mongo_stop` to turn off the db and `mongo_start` to bring it back up again. Try the same tests for the si and production servers. NOTE - use `mongo_stop` to stop the current container and NOT remove the data. The tests on the db check if it is up and contains at least one item. Killing the mongodb container (`mongo_drop`) will remove the entries so will cause the service to look failed if launched again.

Extension Tasks

– Integrate the pipeline into Slack for notifications on build failures.

Comments

There are no comments.