

# CheatSheet: Concourse

## VMWARE

```
#+BEGIN_EXPORT HTML <a href="https://github.com/dennyzhang/cheatsheet.dennyzhang.com/tree/master/cheatsheet-concourse-A4"></a> <div id="the whole thing" style="overflow: hidden;"> <div style="float: left; padding: 5px"> <a href="https://www.linkedin.com/in/dennyzhang001"></a></div> <div style="float: left; padding: 5px"><a href="https://github.com/dennyzhang"></a></div> <div style="float: left; padding: 5px"><a href="https://www.dennyzhang.com/slack" target="_blank" rel="nofollow"></a></div> </div>
<br/><br/> <a href="http://makeapullrequest.com" target="_blank" rel="nofollow"></a> #+END_HTML
```

- PDF Link: [cheatsheet-concourse-A4.pdf](#), Category: vmware
- Blog URL: <https://cheatsheet.dennyzhang.com/cheatsheet-concourse-A4>
- Related posts: Bosh CheatSheet, PKS CheatSheet, #denny-cheatsheets

File me Issues or star this repo

## 1.1 Concourse Concepts

Name	Summary
fly	Pipelines are configured entirely via fly CLI
Teams	Every pipeline and one-off build belongs to a team.
Targets	Alias of Concourse endpoints
Pipelines	A pipeline is the result of configuring Jobs and Resources together.
Jobs	Jobs determine the actions of your pipeline,
Resource	Allow you to define inputs and outputs to jobs in a pipeline
Tasks	The smallest configurable unit in a Concourse pipeline is a single task.
Step: tags	Use tag to match workers
bbl (bosh-bootloader)	Command line utility for standing up a BOSH director on any IAAS
Reference	Link: Concourse Concepts, Link: <a href="#">concoursetutorial.com</a>

## 1.2 Concourse Resources & Examples

Name	Summary
Hello world pipeline	example: <a href="#">helloworld.yml</a>
Pass output to input across different tasks	example: <a href="#">tasks-output-input.yml</a>
Trigger pipeline test every 5 minutes	example: <a href="#">run-test-regularly.yml</a>
git-resource	example: <a href="#">example-git.yml</a>
concourse-curl-resource: http download link	example: <a href="#">url-download.yml</a>
file-downloader-resource	enable downloading pivotal product files and stemcells
github-release	
docker-image-resource: docker image	
gcs-resource: google bucket	
Reference	GitHub: <a href="#">pcf-pipelines</a>

## 1.3 Concourse Containers

Name	Summary
Login to container	<code>fly -t \$target hijack -u \$job_url</code>
Run command without sudden exit	<code>nohup SOMECOMMAND &amp;; tail -f nohup.out</code>
Hijack to containers with a successful run	By design when task has passed, the container will be deleted

## 1.4 Concourse Pipeline

Name	Summary
List all authenticated end points	<code>fly targets</code>
Get fly version	<code>fly --version</code>
fly login	<code>fly -t \$target login -c \$concourse_url -n \$team_name</code>
Get pipeline definition	<code>fly -t \$target get-pipeline -p \$job-name &gt; job.yaml</code>
Update pipeline	<code>fly -t \$target set-pipeline -p \$pipeline_name -c pipeline.yml -l credentials.yml</code>
Update pipeline	<code>fly -t \$target set-pipeline -p \$pipeline_name -c pipeline.yml -v env1=var1</code>
Destroy pipeline	<code>fly -t \$target destroy-pipeline -p \$pipeline_name</code>
Format pipeline	<code>fly format-pipeline -c pipeline.yml</code>
Validate pipeline	<code>fly validate-pipeline -c pipeline.yml</code>

<https://raw.githubusercontent.com/dennyzhang/cheatsheet.dennyzhang.com/master/cheatsheet-concourse-A4/concourse-sample.png>

## 1.5 Concourse Resource

Name	Summary
github-release resource get the old tag	<code>fly -t \$target check-resource -r \$pipeline-name/\$resource-name -f tag</code>
github release use a specific tag	<code>github-release-specific-tag.md</code>
bosh-io-stemcell resource get the old version	<code>fly -t \$target check-resource -r \$pipeline-name/ubuntu-xenial-stemcel</code>

## 1.6 Concourse Security

Name	Summary
Fly login to the default main team	<code>fly -t \$target login -n main -c \$concourse_url</code>
Fly login to a different team	<code>fly -t \$target login -n \$team_name -c \$concourse_url</code>
List all teams	<code>fly -t \$target teams, fly -t \$target teams -d</code>
Fly logout all targets	<code>fly logout --all</code>
Install ansible vault in mac	<code>brew install ansible</code>
Create a team	<code>fly -t \$target set-team -n \$team_name --allow-all-users</code>
Destroy a team	<code>fly -t \$target destroy-team --team-name \$team_name</code>
Grant Github team access	<code>fly -t \$target set-team -n \$team_name --github-org \$my_org</code>
Login to a team	<code>fly -t \$target set-team -n \$team_name</code>
Fly conf file	<code>~/.flyrc</code>
Reference	After login to the <b>main</b> team, you can update other teams. Link: <a href="#">Concourse Team</a>

## 1.7 Concourse Workers

Name	Summary
List all concourse workers	<code>fly -t \$target workers</code>
List all concourse containers	<code>fly -t \$target containers</code>

<https://raw.githubusercontent.com/dennyzhang/cheatsheet.dennyzhang.com/master/cheatsheet-concourse-A4/concourse-design.png>

## 1.8 fly Online Usage

```
bash-3.2$ fly --help
error: Usage:
  fly [OPTIONS] <command>
```

### Application Options:

<code>-t, --target=</code>	Concourse target name
<code>-v, --version</code>	Print the version of Fly and exit
<code>--verbose</code>	Print API requests and responses
<code>--print-table-headers</code>	Print table headers even for redirected output

## Help Options:

-h, --help                      Show this help message

## Available commands:

abort-build	Abort a build (aliases: ab)
builds	List builds data (aliases: bs)
check-resource	Check a resource (aliases: cr)
checklist	Print a Checkfile of the given pipeline (aliases: cl)
containers	Print the active containers (aliases: cs)
destroy-pipeline	Destroy a pipeline (aliases: dp)
destroy-team	Destroy a team and delete all of its data (aliases: dt)
execute	Execute a one-off build using local bits (aliases: e)
expose-pipeline	Make a pipeline publicly viewable (aliases: ep)
format-pipeline	Format a pipeline config (aliases: fp)
get-pipeline	Get a pipeline's current configuration (aliases: gp)
help	Print this help message
hide-pipeline	Hide a pipeline from the public (aliases: hp)
hijack	Execute a command in a container (aliases: intercept, i)
jobs	List the jobs in the pipelines (aliases: js)
login	Authenticate with the target (aliases: l)
logout	Release authentication with the target (aliases: o)
order-pipelines	Orders pipelines (aliases: op)
pause-job	Pause a job (aliases: pj)
pause-pipeline	Pause a pipeline (aliases: pp)
pause-resource	Pause a resource (aliases: pr)
pipelines	List the configured pipelines (aliases: ps)
prune-worker	Prune a stalled, landing, landed, or retiring worker (aliases: pw)
rename-pipeline	Rename a pipeline (aliases: rp)
rename-team	Rename a team (aliases: rt)
set-pipeline	Create or update a pipeline's configuration (aliases: sp)
set-team	Create or modify a team to have the given credentials (aliases: st)
status	Login status
sync	Download and replace the current fly from the target (aliases: s)
targets	List saved targets (aliases: ts)
teams	List the configured teams (aliases: t)
trigger-job	Start a job in a pipeline (aliases: tj)
unpause-job	Unpause a job (aliases: uj)
unpause-pipeline	Un-pause a pipeline (aliases: up)
unpause-resource	Unpause a resource (aliases: ur)
validate-pipeline	Validate a pipeline config (aliases: vp)
volumes	List the active volumes (aliases: vs)
watch	Stream a build's output (aliases: w)
workers	List the registered workers (aliases: ws)

## 1.9 fly set-pipeline Online Usage

```
> fly set-pipeline --help
```

error: Usage:

```
fly [OPTIONS] set-pipeline [set-pipeline-OPTIONS]
```

## Application Options:

-t, --target=	Concourse target name
-v, --version	Print the version of Fly and exit
--verbose	Print API requests and responses
--print-table-headers	Print table headers even for redirected output

## Help Options:

-h, --help                      Show this help message

**[set-pipeline command options]**

<code>-n, --non-interactive</code>	Skips interactions, uses default values
<code>--no-color</code>	Disable color output
<code>--check-creds</code>	Validate credential variables against credential manager
<code>-p, --pipeline=</code>	Pipeline to configure
<code>-c, --config=</code>	Pipeline configuration file
<code>-v, --var=[NAME=STRING]</code>	Specify a string value to set for a variable in the pipeline
<code>-y, --yaml-var=[NAME=YAML]</code>	Specify a YAML value to set for a variable in the pipeline
<code>-l, --load-vars-from=</code>	Variable flag that can be used for filling in template values in configuration

**1.10 More Resources**

<http://www.mikeball.info/blog/concourse-git-resource/>

<https://concoursetutorial.com/>

License: Code is licensed under MIT License. `#+BEGIN_EXPORT HTML` `<a href="https://www.dennyzhang.com"></a> <a href="https://www.dennyzhang.com"></a>`  
`<a href="https://www.linkedin.com/in/dennyzhang001"></a> <a href="https://github.com/dennyzhang"></a> <a href="https://www.dennyzhang.com/slack" target="_blank" rel="nofollow"></a> #+END_HTML`