```python
# Universal Docker Setup Tool by Raphael Lucas Dittberner
import os
import subprocess
import sys
import typer
import yaml

app = typer.Typer()

if "REPO" in os.environ:
    repo = os.environ['REPO'] + '/'
    print("Your current repository path is " + repo + ". To change it, change the REPO environment variable.")
else:
    repo = ''
    print("The tool will use the config file from the directory it is running in. If you want to change this, "
          "set the REPO environment variable.")


config = repo + 'udtconfig.yml'
with open(config, 'r') as udtcfg:
    read_config = yaml.safe_load(udtcfg)

compose_config = repo + "docker-compose.yml"

services = read_config['services']


@app.command(help="Create docker containers for services specified in utdconfig.yml without starting them")
def create():
    subprocess.run(["rm", "-rf", compose_config])
    invalid_options = ['addons', 'bindings', 'commands']
    with open(config) as oldfile, open(compose_config, 'w') as newfile:
        for line in oldfile:
            if not any(invalid_option in line for invalid_option in invalid_options):
                newfile.write(line)
    subprocess.run(["docker-compose", "-f", compose_config, "up", "--no-start"])
    for i in list(read_config['services']):
        for index in list(read_config['services'][i]['addons']['bindings']['commands']):
            subprocess.run([index])


@app.command(help="Start all or specific services specified in udtconfig.yml")
def start():
    start_service = input("Enter the name of the service you want to start, or press ENTER to start all: ")
    if start_service == "":
        start_service = list(services)
        index = 0
        for i in start_service:
            subprocess.run(["docker-compose", "start", start_service[index]])
            index += 1
        sys.exit(0)
    else:
        subprocess.run(["docker-compose", "start", start_service])
        sys.exit(0)


@app.command(help="Stop all or specific services specified in udtconfig.yml")
def stop():
    stop_service = input("Enter the name of the service you want to stop, or press ENTER to stop all: ")
```

```
    if stop_service == "":
        while True:
            confirm = input("Are you sure you want to stop all services? Type YES or NO! ")
            if confirm == "NO":
                print("aborting")
                sys.exit(0)
            elif confirm == "YES":
                stop_service = list(services)
                index = 0
                for i in stop_service:
                    subprocess.run(["docker-compose", "stop", stop_service[index]])
                    index += 1
                sys.exit(0)
            else:
                print("Please type YES or NO!")
                continue
    else:
        subprocess.run(["docker-compose", "stop", stop_service])
        sys.exit(0)


@app.command(help="Run shell in docker container of specified service")
def shell():
    service = input("Enter service name: ")
    subprocess.run(["docker", "compose", "run", service, "sh"])


@app.command(help="Run karaf shell in specified openhab container")
def karaf():
    service = input("Enter service name: ")
    subprocess.run(["docker", "compose", "run", service, "/openhab/runtime/bin/client"])


@app.command(help="Stop all services and delete all docker containers")
def down():
    subprocess.run(["docker-compose", "-f", compose_config, "down"])
    sys.exit(0)


if __name__ == "__main__":
    app()
```