

[Open in app](#)[Get started](#)Zhimin Wen · [Follow](#)

Sep 14, 2019 · 2 min read



Setup Local DNS Server on Macbook

When testing cloud technologies, frequently we are required to point to some DNS entries or even some wildcard DNS names. We commonly see these in Cloudfoundry, Openshift and so on.

On the server-side, we can set up the DNS server to resolve the requirement. On the client-side, the DevOps' laptop, a short-term solution is to add those wildcard entries into the `/etc/hosts`. However, probably a better approach is to set up a local DNS server to address the issue.

Thanks to the Linux flavor of Mac OS, the once looks like daunting work is actually pretty straight-forward. We will use the lightweight [dnsmasq](#), which provides a lightweight local DNS server and a DNS forwarder among many other features. This paper just documents the steps to set up a local DNS server based on the well-spread information and testing.

First, install dnsmasq with brew

```
brew install dnsmasq
```

Secondly, update the configuration file to enable the configuration from the specified directory

```
cp /usr/local/etc/dnsmasq.conf /usr/local/etc/dnsmasq.conf.orig  
echo "conf-dir=/usr/local/etc/dnsmasq.d/,*.conf" | tee  
/usr/local/etc/dnsmasq.conf
```

Create a “conf” file for the required domain resolving in the defined directory, for an



[Open in app](#)[Get started](#)

```
address=/master.poc.ocp.io.local/192.168.20.21
address=/worker1.poc.ocp.io.local/192.168.20.22
address=/worker2.poc.ocp.io.local/192.168.20.23
address=/.apps.poc.ocp.io.local/192.168.20.21
```

Notice the last one is the wildcard dns entry.

Now we need to instruct Mac to resolve the DNS with the dnsmasq. Other than changing the system DNS server, we enable the additional DNS resolver by creating the directory,

```
sudo mkdir -p /etc/resolver
```

Create a file named with the domain name to be resolved, and inside the file just define the nameserver to localhost, as shown below

```
cat /etc/resolver/poc.ocp.io.local
nameserver 127.0.0.1
```

Lastly, start the dnsmasq service

```
sudo brew services start dnsmasq
```

Testing the DNS record.

```
dig master.poc.ocp.io.local @localhost +short
192.168.20.21
```

```
dig any.apps.poc.ocp.io.local @localhost +short
192.168.20.21
```



[Open in app](#)[Get started](#)

Tips:

You may need to clear the DNS Cache to let the change take effect.

```
sudo dscacheutil -flushcache  
sudo killall -HUP mDNSResponder
```

