Running Puppet Master and Puppet Agent in Docker on the same Windows machine requires Docker Desktop with Linux containers enabled. Below is a step-by-step guide to setting up Puppet in Docker.

Prerequisites

1. Install Docker Desktop on Windows

- Download from: [Docker Official Website](https://www.docker.com/products/docker-desktop)

- Enable WSL 2 backend (Recommended for performance).

- Ensure Docker is running in Linux Containers Mode.

2. Verify Docker Installation

Run the following command in PowerShell or Command Prompt:

powershell

docker --version

It should return the installed Docker version.

Step 1: Pull Puppet Docker Images

Puppet provides official Docker images. Pull the latest images for Puppet Server (Master) and Puppet Agent.

powershell

docker pull puppet/puppetserver

docker pull puppet/puppet-agent

Step 2: Run Puppet Master in a Container

Start the Puppet Master container:

powershell

docker run -d --name puppet-master `

--hostname puppet `

-p 8140:8140 `

puppet/puppetserver

#Explanation:

- `-d` → Runs the container in the background.

- `--name puppet-master` → Assigns a name to the container.

- `--hostname puppet` → Sets the internal hostname to `puppet`.

- `-p 8140:8140` → Maps Puppet's default port (8140) from container to host.

- `puppet/puppetserver` → Specifies the Puppet server image.

Verify that the Puppet Server is running:

powershell

docker ps

You should see `puppet/puppetserver` in the list.

Step 3: Run Puppet Agent in a Container

Start the Puppet Agent container and connect it to the Puppet Master:

powershell

docker run -d --name puppet-agent `

--hostname puppet-agent `

--add-host puppet:host-gateway `

puppet/puppet-agent agent --server=puppet

#Explanation:

- `--hostname puppet-agent` → Sets hostname for the agent.

- `--add-host puppet:host-gateway` → Allows the agent to reach the Puppet Master running on the same machine.

- `puppet/puppet-agent agent --server=puppet` → Starts Puppet Agent and points it to the Puppet Master.

Check running containers:

powershell

docker ps

Step 4: Configure SSL Certificates

Puppet Agent will generate a certificate request that must be signed by Puppet Master.

#1. Check Pending Certificate on Puppet Master

powershell

docker exec -it puppet-master puppetserver ca list

#2. Sign the Certificate

powershell

docker exec -it puppet-master puppetserver ca sign --all

#3. Test Puppet Agent Communication

Now, manually trigger Puppet Agent to check for configurations:

powershell

docker exec -it puppet-agent puppet agent --test

If everything is correct, the agent should successfully communicate with the master.

Step 5: Create and Apply a Puppet Manifest

A Puppet Manifest defines the desired configuration for the agent.

#1. Create a Manifest on the Puppet Master

powershell

docker exec -it puppet-master bash

Inside the container, create a manifest file:

bash

mkdir -p /etc/puppetlabs/code/environments/production/manifests

nano /etc/puppetlabs/code/environments/production/manifests/site.pp

Add the following:

puppet

node 'puppet-agent' {

package { 'vim':

ensure => installed,

}

}

Save the file (`CTRL+X`, `Y`, `Enter`).

#2. Apply Manifest on Puppet Agent

Run:

powershell

docker exec -it puppet-agent puppet agent --test

The agent should now install `vim` inside its container.

Step 6: Automate Puppet Runs

To ensure Puppet Agent runs automatically every 30 minutes, enable the background service:

powershell

docker exec -it puppet-agent puppet resource service puppet ensure=running enable=true

Step 7: Verify Setup

Check the logs of Puppet Server:

powershell

docker logs puppet-master

Check the logs of Puppet Agent:

powershell

docker logs puppet-agent

Cleanup (Optional)

If you want to remove everything:

powershell

docker stop puppet-master puppet-agent

docker rm puppet-master puppet-agent

docker rmi puppet/puppetserver puppet/puppet-agent

Conclusion

You have successfully set up Puppet Master and Agent inside Docker containers on Windows. 🚀 Now, you can use Puppet to automate system configurations inside containers.