bash script to build dns server and map a domain to another server

Here is a simple bash script to build a DNS server using BIND and map a domain to another server:

bash

```
#!/bin/bash

# Install BIND
sudo apt-get update
sudo apt-get install bind9

# Create DNS zone file
sudo nano /etc/bind/db.example.com

# Add the following contents to the file
$TTL    604800
@       IN      SOA     ns1.example.com. admin.example.com. (
                  1         ; Serial
             604800         ; Refresh
              86400         ; Retry
            2419200         ; Expire
             604800 )       ; Negative Cache TTL
;
@       IN      NS      ns1.example.com.
@       IN      A       192.168.1.2
www     IN      CNAME   server.example.com.

# Save and close the file

# Configure BIND
sudo nano /etc/bind/named.conf.local
```

```
# Add the following contents to the file
zone "example.com" {
    type master;
    file "/etc/bind/db.example.com";
};

# Save and close the file

# Restart BIND
sudo systemctl restart bind9

# Map domain to another server
sudo nano /etc/hosts

# Add the following line to the file
192.168.1.3 server.example.com

# Save and close the file
```

Replace "example.com" with your domain name, "192.168.1.2" with the IP address of your DNS server, and "192.168.1.3" with the IP address of the server you want to map your domain to.

After running this script, your DNS server should be set up and your domain should be mapped to the specified server.

make dns server using python

Python has a number of libraries available for DNS, including dnspython and pydns. Here is a simple example using dnspython to create a DNS server that responds to DNS queries:

python

```
import dns.message
import dns.query
import dns.resolver
import dns.rdatatype
import socketserver

class DNSHandler(socketserver.BaseRequestHandler):
    def handle(self):
        data = self.request[0]
        socket = self.request[1]
        dns_request = dns.message.from_wire(data)

        # Extract the requested domain name and type
        question = dns_request.question[0]
        domain_name = question.name.to_text()
        record_type = question.rdtype

        # Look up the domain name in the DNS resolver
        resolver = dns.resolver.Resolver()
        response = resolver.query(domain_name, record_type)

        # Build a DNS response
        dns_response = dns.message.make_response(dns_request)
        dns_response.answer = response.rrset

        # Send the DNS response
        socket.sendto(dns_response.to_wire(), self.client_address)

if __name__ == "__main__":
    # Set up the DNS server
    dns_server = socketserver.UDPServer(('localhost', 53), DNSHandler)

    # Serve DNS requests indefinitely
    dns_server.serve_forever()
```

This creates a UDP server listening on port 53 for DNS requests. When a request is received, it extracts the requested domain name and type, looks up the domain name using the DNS resolver, and builds and sends a DNS response. Note that this example is for educational purposes only and should not be used in a production environment without further modification and testing.

make a dns server and map another domain to my apache server

Here is an example of how to set up a DNS server using BIND and map another domain to an Apache server:

1. Install BIND

sql
```
sudo apt-get update
sudo apt-get install bind9
```

2. Configure BIND

bash
```
sudo nano /etc/bind/named.conf.local
```

Add the following contents to the file:

bash
```
zone "example.com" {
    type master;
    file "/etc/bind/db.example.com";
};
```

Save and close the file.

3. Create DNS zone file

bash
```
sudo nano /etc/bind/db.example.com
```

Add the following contents to the file:

less

```
$TTL    604800
@       IN      SOA     ns1.example.com. admin.example.com. (
                  1         ; Serial
             604800         ; Refresh
              86400         ; Retry
            2419200         ; Expire
             604800 )       ; Negative Cache TTL
;
@       IN      NS      ns1.example.com.
@       IN      A       <IP address of your DNS server>
www     IN      A       <IP address of your Apache server>
```

Replace `example.com` with your domain name, `<IP address of your DNS server>` with the IP address of your DNS server, and `<IP address of your Apache server>` with the IP address of your Apache server.

Save and close the file.

4. Restart BIND

```
sudo systemctl restart bind9
```

5. Map another domain to your Apache server

You will need to add a VirtualHost configuration to your Apache server to map the domain to your Apache server. Here is an example configuration:

bash

```
sudo nano /etc/apache2/sites-available/anotherdomain.com.conf
```

Add the following contents to the file:

less

```
<VirtualHost *:80>
    ServerName anotherdomain.com
    DocumentRoot /var/www/html

    <Directory /var/www/html>
        Options FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

Replace `anotherdomain.com` with the domain name you want to map to your Apache server. Save and close the file.

6. Enable the new virtual host and restart Apache

```
sudo a2ensite anotherdomain.com.conf
sudo systemctl reload apache2
```

Your DNS server should now be set up and another domain should be mapped to your Apache server. Note that it may take some time for DNS changes to propagate, so you may need to wait a bit before testing the new domain mapping.

ChatGPT Mar 14 Version