

Práctico FCEFyN Redes de computadoras

Trabajo Práctico 7

Docente: Matías R. Cuenca del Rey

Mail: mcuenca@unc.edu.ar

Ayudantes alumnos: Elisabeth Leonhardt - Andrés Serjoy

Redes de computadoras

Facultad de Ciencias Exactas, Físicas y Naturales

Universidad Nacional de Córdoba

Práctico 7: Aplicación.

Presentación de consignas.

Ejercicio 1: Ruteo internet.

Recomendaciones

- Lea con cuidado las consignas.
- Tenga certeza de los comandos que ejecuta.
- Para contenerización utilizar Docker CE.

Esquema

- Se realizará bajo IPv6. No se usará IPv4.
- Se usarán dos máquinas físicas por grupo de 4 personas.
- Se usará la máquina virtual Desktop.
- Se usarán containers dentro de la máquina virtual para simular routers.
- Se usarán containers dentro de la máquina virtual para los distintos servicios.
- Se usarán containers dentro de la máquina virtual para simular los hosts.
- Cada container podrá almacenar un único servicio.

Diagrama

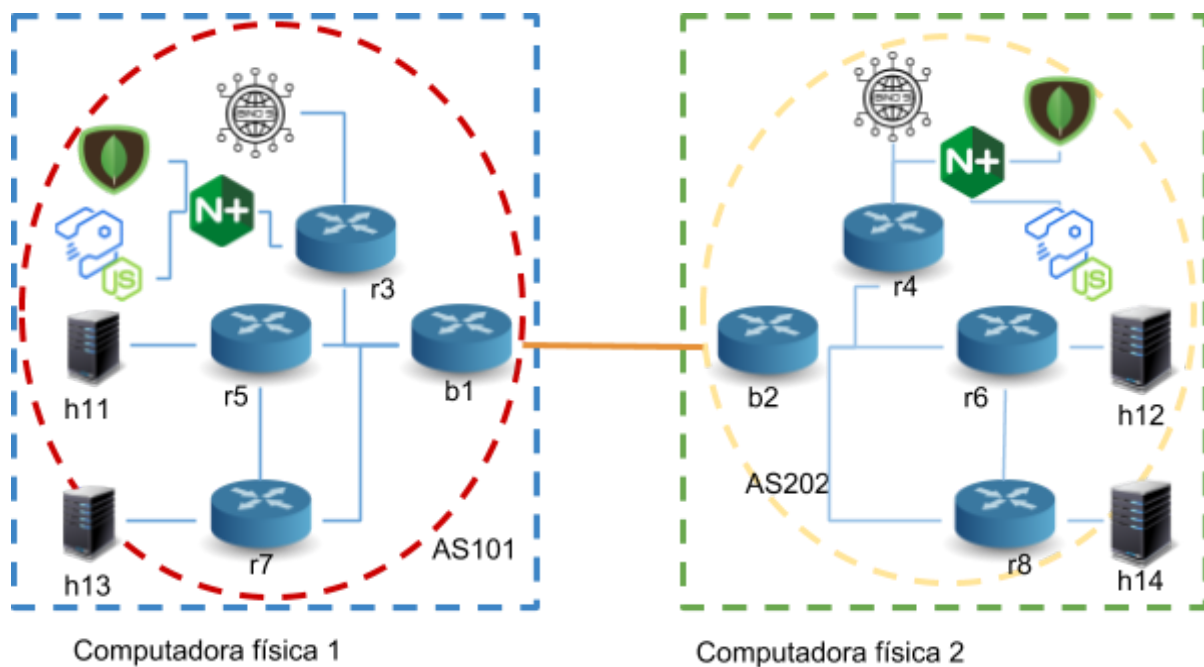


Tabla de asignación de direcciones IPv6

Crear la tabla de asignación de direcciones IP

Computadora	Interfaz de red	Dirección IP
-------------	-----------------	--------------

Links de ayuda

Instalación de Docker CE: <https://docs.docker.com/install/linux/docker-ce/ubuntu/>

Instalación de Docker compose:

<https://www.digitalocean.com/community/tutorials/how-to-install-docker-compose-on-ubuntu-16-04>

Quagga y otros software OpenSource para ruteo:

<https://keepingitclassless.net/2015/05/open-source-routing-comparison/>

Strapi: <https://hub.docker.com/r/strapi/strapi/>

Strapi + Nginx: <https://blog.strapi.io/how-to-deploy-a-strapi-application/>

Consignas

Configuración de Red

- 1.- Asignar redes IPv6 para cada sistema autónomo.
- 2.- Asignar un nombre de dominio único para cada sistema autónomo.
- 3.- Completar la tabla de asignación de redes.
- 4.- Configurar los sistemas autónomos con BGP.
- 5.- Interconectar los sistemas autónomos.
- 6.- Configurar OSPFv3 como protocolo de ruteo interno dentro de cada sistema autónomo

DNS

- 7.- Configurar DNS en cada sistema autónomo y asignar registro AAAA y PTR a cada dirección IP asignada.
- 8.- Configurar la solución de manera tal que desde cualquier host sea posible resolver cualquier nombre de dominio.

Aplicación Web

- 9.- Implementar el stack NGINX + NodeJS + MongoDB utilizando Strapi como web framework.
- 10.- MongoDB debe tener una configuración de alta disponibilidad con 3 servicios corriendo.
- 11.- Crear una API distinta en cada sistema autónomo. Para el sistema autónomo 101, crear un Content-type de "Inventario de Servidores". Para el sistema autónomo 202, crear un Content-type de "Clientes de servicios de proxy"
- 12.- Comprobar que es posible Crear, Modificar y Eliminar entradas en estas aplicaciones web a través de llamadas a las API utilizando Autenticación.
- 13.- Configurar NGINX para soportar también TLS. Es decir, la aplicación web debe funcionar en puerto 80 y puerto 443. No es necesario usar NGINX para balancear mongo.
- 14.- Configurar los hosts para que reconozcan los certificados configurados en el punto anterior como válidos.

Proxy HTTP

15.- Implementar un proxy HTTP en cada sistema autónomo que pueda ser usado por los Hosts.