**Networking**

1. **What is something generally hosted behind port 80? Why might you switch it to be**

**behind 8080?**

Answer:

Ports

Well-known/standard ports

Private Port

User Ports

1024–49151

Above 49151

0-1023

HTTP (Communication)

Runs over TCP/IP (connection)

Additional (may be for Proxy)

Default

80 (Webserver)

8080

http://www.prajith.com:8080

http://www.prajith.com

Port 80 is the default HTTP port and behind Port 80, usually a webserver is hosted. Hence, anyone from internet who wants to access my website www.prajith.com will be by default landed on port 80.

Port 8080 is the alternative port standard for any other HTTP service, or for that matter, any other alternative webserver. If someone from internet wants to access the service running on 8080, he/she will have to explicitly mention the same on the URL.

Below are the reasons one would like to host a webserver on port 8080:

1. An additional webserver is needed and port 80 is already occupied.
2. If there is a user that wants an alternative webserver but do not have authority to run services below port 1024.
3. There is a need for a personally hosted webserver.
4. If there is a need for a proxy server, port 8080 is the default.

**2. What can you do with ssh besides using it as an interactive shell?**

**Answer:**

Along with being secure shell, SSH can be used for various other purposes like:

1. Password-less login from one terminal to another.

* Use ssh-keygen to generate public-private key-pair
* Copy the public key to authorized\_key file of the user in the target server

1. Secure copy files between two servers.

*scp path\_to\_sourcefile username@hostname:path\_to\_destination*

1. Run commands/programs on remote host

*ssh user@host << MYHEREDOC*

*Command1*

*Command2*

*Script3*

*MYHEREDOC*

1. Access remote services by redirection from local port to remote port

*ssh user@remote -L 8080:jenkins:8080*

**3. What’s ICMP?**

**Answer:**

ICMP stands for Internet Control Message Protocol.

This protocol is used by networking devices (such as routers) for reporting the error to the sending device if any of its data packet was not received to the receiving device.

This helps the sending device or the upstream router to reduce the rate of transmission.

ICMP is also used in network troubleshooting utilities like traceroute, dig and ping.

**4. What is the most significant difference between IPv4 and IPv6?**

**Answer**

* IPv4 is a 32-bit numeric addressing scheme,
* IPv6 is a 128 bit alpha-numeric addressing scheme.

Due to large address space, IPv6 has a capability of supporting huge range of IP addresss with better complexity and efficiency

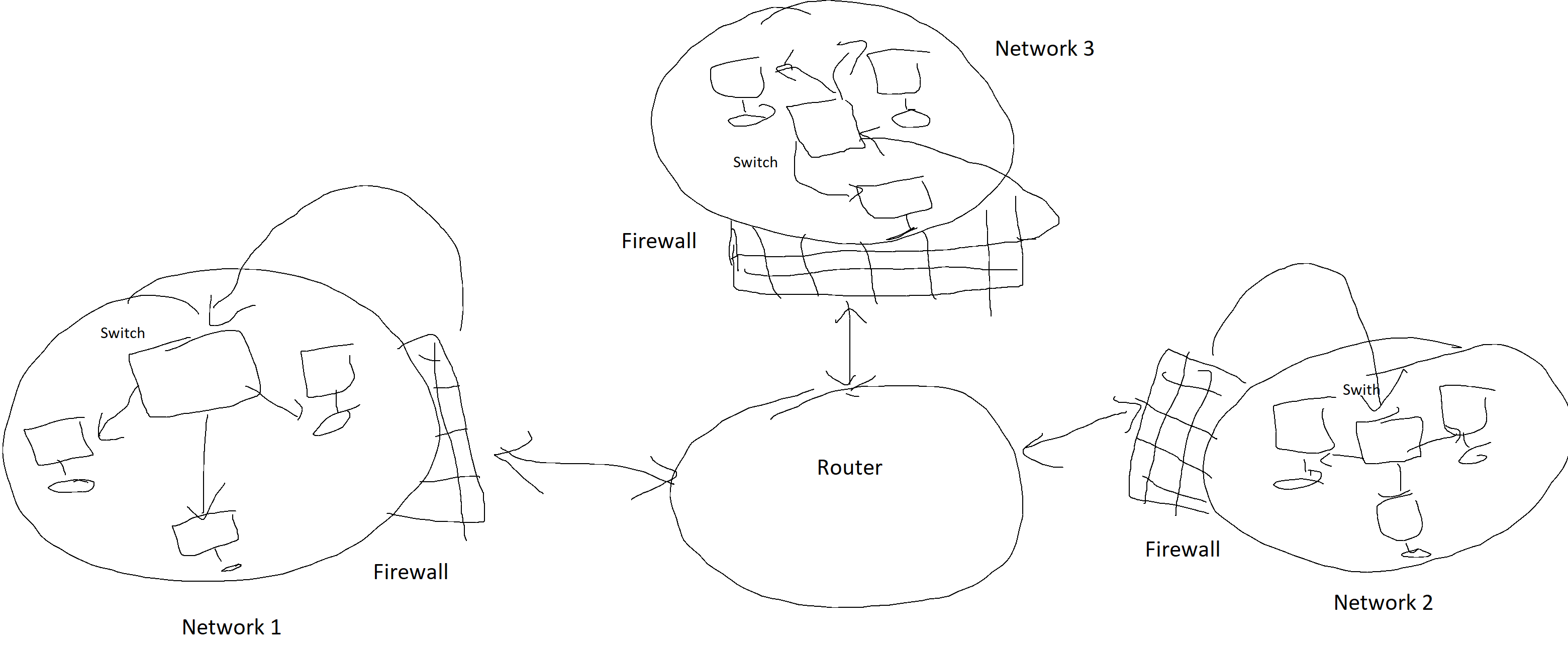
**5. What’s the difference between TCP and UDP protocols?**

**Answer**

|  |  |
| --- | --- |
| **TCP** | **UDP** |
| Targets reliability | Targets speed |
| Connection Oriented, connection is established before communication starts.  (1) Reports error for package loss  (2) Guarantees the delivery. If not done, reattempt | Connection-less, no connection established  and communication starts on best-effort mode (1) No error report for packet loss (2) For lost packet, no recovery |
| Flow-Control | No Flow control |
| Data acknowledgement, if no acknowledgement received within window reattempt to send the packet | No acknowledgement, if a packet is lost, no way to recover |
| Usage where reliability is needed: (1) http/https (2) SMTP (3) SFTP | Usage where speed matters and packet loss does not: (1) VoIP (2) Live text and video streaming  (3) Cloud/Online gaming |

**6. What are the differences between firewall, router and switch?**

**Answer**



|  |  |  |
| --- | --- | --- |
| **Switch** | **Router** | **Firewall** |
| Connects all devices within network | Connects two different networks | Protects the network |
| Setup between 2 nodes within network | Setup at gateway | setup pn network level |
| Works on MAC addresses | Works on public-private IP addresses | Works on network address |

**7. What different ways can you expose something to the public in Kubernetes?**

**Answer** In Kubernetes, applications are exposed using services.  
These can me done with three types of services:

* **NodePort** - Exposes the Service on the same port of each selected Node in the cluster using NAT. Makes a Service accessible from outside the cluster using <NodeIP>:<NodePort>. Superset of ClusterIP.
* **LoadBalancer** - Creates an external load balancer in the current cloud (if supported) and assigns a fixed, external IP to the Service. Superset of NodePort.
* **ExternalName** - Exposes the Service using an arbitrary name (specified by externalName in the spec) by returning a CNAME record with the name. No proxy is used. This type requires v1.7 or higher of kube-dns.

**Linux**

**1. What would you use to test a HTTP-1 connection? What about a HTTP-2 connection?**

**Answer:**

HTTP1 – Can be tested for different scenarios using below methods  
 (a) TRACE - allows the client to see what is being received at the other end of the request chain and use that data for testing or diagnostic information

### (b) HEAD - This method is often used for testing hypertext links for validity, accessibility, and recent modification.

### HTTP2 – Testing HTTP-2 can be done using various tools available such as:

### Browser indicators

### Command line tools such as is-http and h2i

**2. What’s the name of the protocol responsible for keeping Linux time updated?**

**Answer**

Network Time Protocol (NTP), that with the help if a daemon program (ntpd) polls the public time servers to update the server time.

If polling a public time server is not viable, the LAN NTP server can also be setup.

**3. How would you go about obtaining log output from a service on a modern systemd**

**based Linux distribution?**

**Answer**

In modern Linux distributions, *systemd* is the standard for system management and centralized logging. The journal component aka journal daemon (*journald*) of system helps with this.

In order to scrutinize the logs generated by *journald*, the *journalctl* utility is used.

In the unit-based scheme of *journalctl*, a service can be considered as a unit and in order to see logs related to a service (say Jenkins), below command can be used:

*journalctl -u jenkins.service*

**4. Which commands would you use to check:**

**a. IP address and subnet mask of all network interfaces**

**Command**:

*ifconfig -a | cut -d ':' -f 1 | grep -B1 netmask*

**Output**:

*eth0*

*inet 172.31.26.73 netmask 255.255.240.0 broadcast 172.31.31.255*

*--*

*lo*

*inet 127.0.0.1 netmask 255.0.0.0*

**Explanation**

* ifconfig -a will return ip info about all network interfaces
* cut command will cut the lines at ‘:” and will take the first field
* In the output so far, grep -B1 netmask will find the lines with string “netmask” and also one line Before (that is the name of the interphase.

**b. Routing table**

**Answer**

Below is the command I will use to display the routing table:

[ec2-user@ip-172-31-26-73 ~]$ netstat -r

Kernel IP routing table

Destination Gateway Genmask Flags MSS Window irtt Iface

default ip-172-31-16-1. 0.0.0.0 UG 0 0 0 eth0

instance-data.u 0.0.0.0 255.255.255.255 UH 0 0 0 eth0

172.31.16.0 0.0.0.0 255.255.240.0 U 0 0 0 eth0

**5. What’s the name of the most commonly used Linux scheduling package?**

Completely fair solution (CFS) being the default schedular for various linux distributions seems to be the one that is most used.  
However, the most popular doesn’t mean the best in all cases 😊

**6. Give a few examples of HTTP servers.**

**Answer**  
Nginx

Apache

Lighttpd

IIS

**AWS**

**1. What is used for management of users and permissions in AWS?**

**Answer**:

AWS IAM, that stands for Identity and Access Management is the service used for users and permission for access to AWS resources.  
**Components**  
Users: Define AWS users  
Group: Define AWS user group

Rules: Defines the level of authorization to use any AWS service  
Policies: Pre-defined access rules that can be attached to the users and groups

**2. What Amazon service would you use as a front end application and why? What are the**

**alternatives?**

**Answer**:  
For front-end development, AWS amplify is the service that can help front-end developers.  
Alternatives

* Cloud Application Front End (From AWS Marketplace)
* AWS Mobile Hub (For Mobile Apps)
* Salesforce Heroku
* Parse

**3. Is a s3 bucket located in eu-west-1 also located in us-east-1?**

**Answer**  
AWS S3 buckets are region specific.

However, they can be accessed from any region.

If the automatic and asynchronous copy of objects between s3 bucket is the need, then ‘Cross-Region-Replication’ will help.

**4. Why would you use AWS Lambda instead of running a script directly on a server?**

**Answer**  
AWS Lambda service implements serverless computing that runs the user code based on some event and manages all the resources needed for that execution on its own.

**Advantages of such approach are:**

* No server management needed from user side. AWS Lambda takes care of it.
* User is free from planning for scaling. AWS Lambda takes care of scale up and scale down based on the number if events occurring.
* Planning for high availability is the responsibility of AWS Lambda
* No capacity wastage as AWS Lambda uses resources in the most efficient way.

**5. Why would you use RDS over a traditional DB on an instance? Why would you not use**

**RDS over a traditional db on an instance?**

**Answer  
When to use RDS over a traditional DB on an instance:**

* When Performance is a vital consideration.
* When auto-configuration of High Availability and failover support is an area to focus
* When automated planning for Backup and Restore is an important business need.
* When the simplicity of scaling eases your life.

**When not to use RDS over a traditional DB on an instance:**

There are certain features that are unavailable or not allowed in Amazon RDS.

* When you need Global Transaction IDs, Transportable Tablespace, Replication Filters, and Semi-Synchronous Replication.
* When you need Password and Authentication strengthen plugins.

**6. Why would you choose EKS over a cluster spun up directly in EC2? Why would you not?**

**Answer**

**Reasons to choose EKS**

* It integrates well with AWS resources and services. If most of the operations are on AWS cloud, then EKS is a good choice
* When you want to save time of manual resource bill tracking, EKS can help because it uses serverless model
* When you want to automate the control plane management, you can delicate it to EKS.

**Programming languages & general knowledge**

**1. What’s the name of the package manager used by NodeJS?**

Answer  
NPM is the package manager for NodeJS applications

It gets installed along with Nodejs installation.

**2. What command would you use to install Ruby modules?**

[sudo] gem install package-name

Eg: sudo gem install rails\_utils

**3. What is a RegEx? Where would you use one?**

RegEx are special purpose expressions that are used to match or search a pattern

For instance, if we want to fetch all email IDs from a file then in that case, the regEx will be like  
[any set of anlpha-numneric string, not starting with a number]@[alphabetic string].com

Or

[a-zA-Z] [a-zA-Z0-9]\*@[a-zA-Z]+\.com

**Kubernetes**

**1. How would you restart a webservice that’s hosted in Kubernetes?**

**Answer**

Suppose name of my app deployment is myWebservice, then I will do the rolling restart with:

*kubectl rollout restart deployment/myWebapp*

**2. What are the differences between a pod, a container and a service?**

**Answer**

|  |  |  |
| --- | --- | --- |
| **Pod** | **Container** | **Service** |
| The basic unit of deployment in kubernetes | An isolated environment that is based on a dockerfile | A service can be considered as a logical grouping of pods stitched togather with label selector |
| A pod can contain many container | A container is based  on one dockerfile | A service is associated with a deployment that in turn is associated with multiple pods |
| Serves as a unit of replication | Serves as the logic to isolate an environment | Serves as an internal loadbalancer that proxies container for the traffic received |

**3. How would you view pod logs?**

**Answer**

*kubectl describe pods podName*

**4. What is Helm used for? What are the alternatives?**

**Answer**  
Helm is the Kubernetes package manager that with the help of Helm Charts eases the definition, installation and upgrade of applications.

A few of its alternatives are:

* Gitkube
* Skaffold
* Ksonnet
* Metaparticle
* Manual deployments 😊

**5. What is an ingress controller? How does an ingress resource relate to the controller?**

**Answer**

An Ingress controller in a cluster enables Ingress resource functionality.

An ingress exposes HTTP and HTTPS routs from outside the cluster to the services within cluster.

In other words, an ingress resource is definition, and ingress controller is the one that implements that definition.

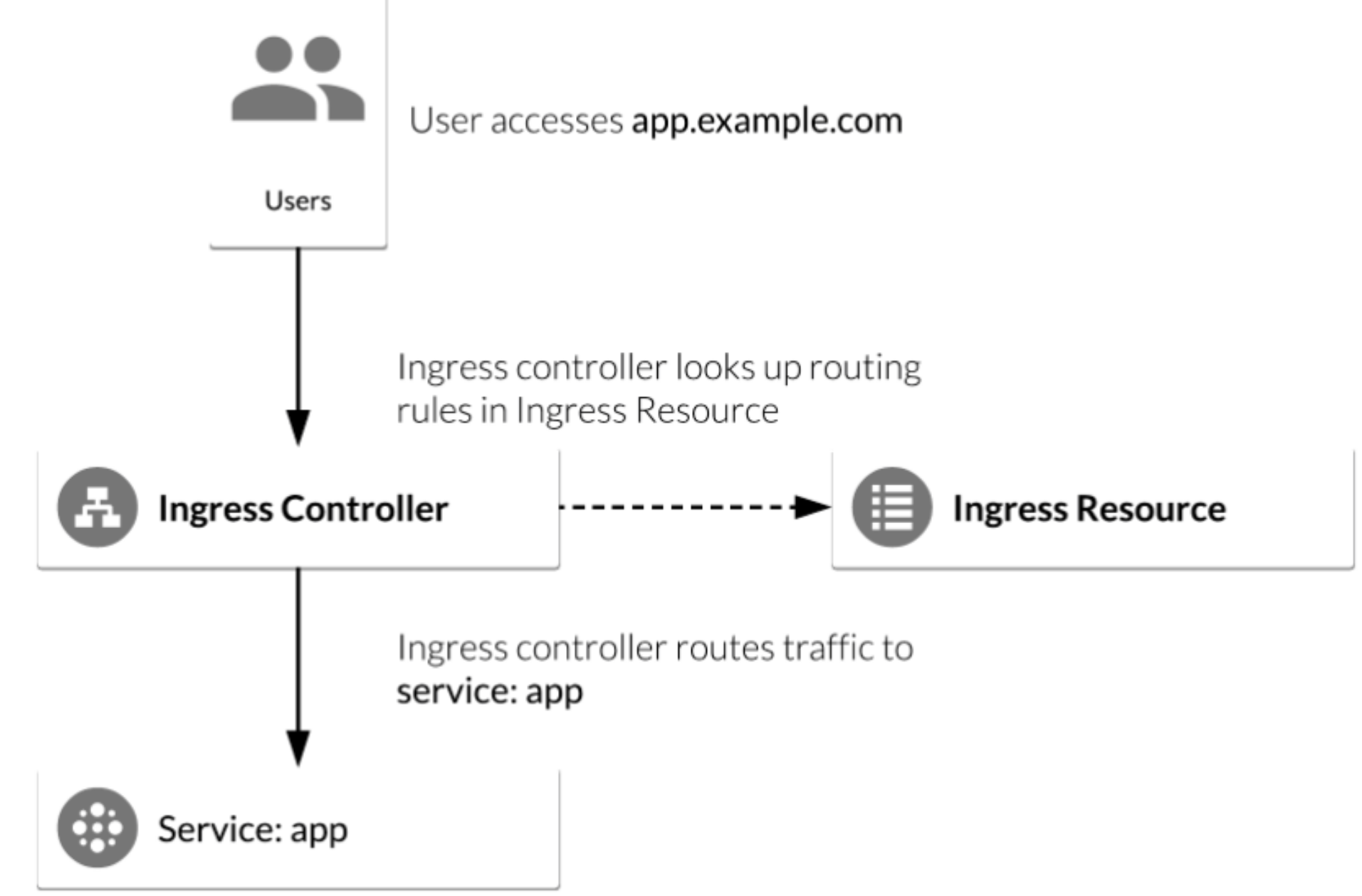


Image resource: <https://www.bogotobogo.com/DevOps/Docker/Docker_Kubernetes_Nginx_Ingress_Controller_GCP_Kubernetes.php>

**6. How would you get into a pod to debug a process?**

**Answer**If the container running inside pod has debugging utilities, we can get into the pod with `exec` command

**Example:**  
To get into a pod running MySql, with a shell connected to terminal:

*Kubectl exec -it MySql –sh*

If the pod container doesn’t have debug utilities, then the approach of ephemeral or sidecar containers can be used.

**Infrastructure as Code**

**1. What is infrastructure as code? What’s the purpose?**

**Answer**Infrastructure as Code is an approach to define the infrastructure in the form of configuration files and code, and saving it in version control. This infrastructure definition may contain – Networks, VM configurations, load balancer and general network topology.

The purpose of Infrastructure as Code is:

* Standardize the configuration so that there is no discrepancy between dev, QA and prod environments.
* Maintain them in version control so that all the configuration changes are always recorded.
* Infrastructure as Code makes the configuration changes faster, less error porone and more convenient.
* It also helps maintain the state and idempotency.

**2. What are some examples of tools you’d use for Infrastructure as code?**

**Answer**  
Tools that can be used for Infrastructure as Code are:

* Ansible
* Chef
* Puppet
* SaltStack
* Vagrant
* Terraform
* Docker (in the form of dockerfiles)
* Cloud services such as:
  + Azure resource manager
  + AWS CloudFormation
  + Google Cloud Deployment Manager