



Free Questions for 300-910 by certsinside

Shared by Willis on 15-04-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

Which approach must be used to integrate DevOps practices into the team structure in an existing organization?

Options:

- A- Create a team that can deliver all the stages with active cooperation between the team members
- B- Create a team that can deliver all the stages with well-defined roles and a communication structure between team members
- C- Create a team for each stage with structured communication channels
- D- Create a team for each stage with active cooperation between the teams

Answer:

A

Question 2

Question Type: MultipleChoice

Refer to the exhibit.

```
<34>1 2020-10-11T22:14:15.003Z 161.20.30.44 su - ID47 - BOM'su root' failed for DevNetUser on /dev/pts/8
```

The text represents a syslog message sent from a Linux server to a centralized log system.

Based on the format of the log message, how must the functionality of the log parser be extended to improve search capabilities?

Options:

- A- Reverse lookup the IP address to add a hostname field
- B- Convert the date to the time zone of the system
- C- Configure the Linux machine to add a UID field to messages
- D- Filter out the text of the message to speed up searches

Answer:

D

Question 3

Question Type: MultipleChoice

Refer to the exhibit.

```
1 ---
2 version: '3'
3 services:
4   database:
5     image: postgres
6     container_name: postgre_data
7     ports:
8       - "5432:5432"
9     environment:
10      POSTGRES_DB: "postgres_database"
11      POSTGRES_USER: "postgres"
12      POSTGRES_PASSWORD: "testpass"
13     networks:
14       - "database_net"
15     healthcheck:
16       test: ['CMD','psql', '--username', 'postgres']
17       interval: 10s
18       timeout: 10s
19       retries: 5
20
21 networks:
22   database_net:
23     driver: "bridge"
```

A docker-compose.yml file implements a postgres database container.

Which .gitlab-ci.yml code block checks the health status of the container and stops the pipeline if the container is unhealthy?

A)

```
Validate Application Infrastructure:
  stage: validate_infrastructure
  before_script:
    - apk add --no-cache docker-compose
  script:
    - docker-compose up
    - sleep 15s
    - health_state = $(docker container logs
      postgres_data | grep healthcheck)
    - $health_state == 'healthy'
```

B)

```
Validate Application Infrastructure:
  stage: validate_infrastructure
  before_script:
    - apk add --no-cache docker-compose
  script:
    - docker-compose up
    - sleep 15s
    - health_state = $(docker container attach
      postgres_data && \
      psql -username postgres status)
    - $health_state == 'running'
```

C)

```
Validate Application Infrastructure:
stage: validate_infrastructure
before_script:
  - apk add --no-cache docker-compose
script:
  - docker-compose up -d
  - sleep 15s
  - health_state=$(docker container inspect
  postgres_data --
  format='{{.State.Health.Status}}')
  - if [[ "$health_state" != "healthy" ]]; then
  exit 1; fi
```

D)

```
Validate Application Infrastructure:
stage: validate_infrastructure
before_script:
  - apk add --no-cache docker-compose
script:
  - docker-compose up
  - sleep 15s
  - health_state = $(docker container state
  postgres --health_status)
  - if [[ "$health_state" != "running" ]]; then
  exit 1; fi
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

Answer:

C

Question 4

Question Type: DragDrop

Get Site Health

Operation Id: `getSiteHealth`

Description: Returns Overall Health information for all sites

GET `/dna/intent/api/v1/site-health`

Request Parameters

Query

timestamp | String

Epoch time(in milliseconds) when the Site Hierarchy is required

Get Overall Network Health

Operation Id: `getOverallNetworkHealth`

Description: Returns Overall Network Health information by Device category (Access, Distribution, Core, Router, Wireless) for any given point of time

GET `/dna/intent/api/v1/network-health`

Request Parameters

Query

timestamp | String

Epoch time(in milliseconds) when the Network health data is

Information from the Cisco DNA Center
drops below 80 and if a network statistic

site and network health on a Cisco DNA

```
params = {}
```

```
params['SITE_HEALTH'] = True
```

```
url = BASE_URL + NETWORK_HEALTH_URL
```

```
params['timestamp'] = timestamp
```



```
BASE_URL = 'https://sandboxnac.cisco.com'
NETWORK_HEALTH_URL = '/dna/intent/api/v1/network-health'
SITE_HEALTH = '/dna/intent/api/v1/site-health'
timestamp = datetime.timestamp()
data = {
    'X-Auth-Token': "asfds"
}
info = {
    [redacted] : timestamp
}
while True:
```

Answer:

Question 5

Question Type: MultipleChoice

Refer to the exhibit

```
[redacted] requests.request('GET', url,
headers=data, [redacted] =info)
if response.json()[0]['accessGoodCount'] < 80:
    trigger_site_alert()
[redacted]
response = requests.request('GET', url,
headers=data, [redacted] =info)
```

params , + SITE_HEALTH HEALTH

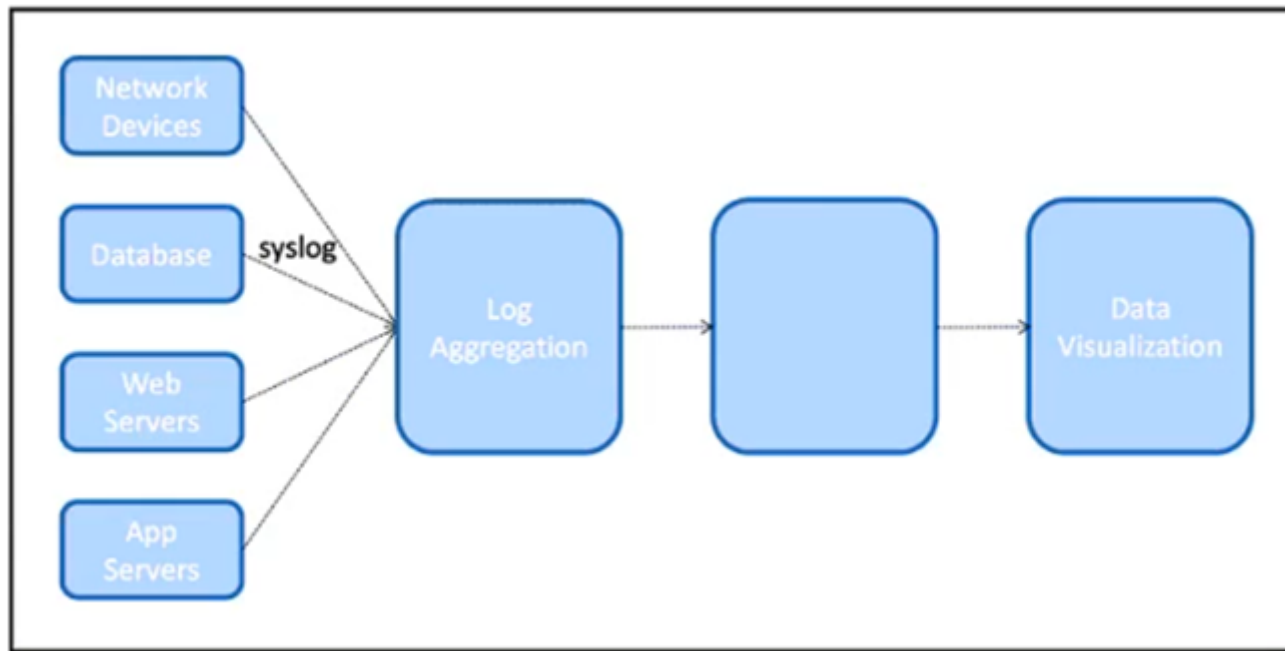
params

url = BASE_URL + NETWORK_HEALTH_URL ,

'query'

'timestamp' info"

'timestamp'



The IT team is creating a new design for a logging system. The system must be able to collect logs from different components of the infrastructure using the SNMP protocol. When the data is collected it will need to be presented in a graphical UI to the NOC team.

What is the architectural component that needs to be placed in the unlabeled box to complete this design?

Options:

A- message queue

B- web server

C- log parser

D- time-series database

Answer:

D

Explanation:

A log parser is a software component that is used to collect and parse log data from various sources, such as infrastructure devices, applications, and services. It is responsible for collecting log data from the various sources and then mapping it to a specific format that can be easily consumed and interpreted by the NOC team. The log parser can also be used to filter out irrelevant data and store the processed data in a time-series database. This data can then be used by the NOC team to generate visualizations and reports, which will help them identify and address any issues within the infrastructure. (Source: Cisco Implementing DevOps Solutions and Practices using Cisco Platforms (DEVOPS) Study Manual Chapter 5, Understanding Logging and Log Analysis)

Question 6

Question Type: MultipleChoice

What is the impact of using the Drone.io CI/CD tool on the local installation step?

Options:

- A- slows down the development
- B- delays the deployment of components
- C- speeds up the procedure
- D- complicates the application process

Answer:

B

Question 7

Question Type: DragDrop

Drag and drop the code from the bottom onto the box where the code is missing to create a Terraform configuration that builds the network environment for a multitier software application. More EPG, Contract, and Filter definitions have been removed from the code.

```
resource "aci_application_profile" "production_multi_app" {
  tenant_dn = aci_tenant.production.tenant.id
  [redacted] = "multi_app_prod"
```

Answer: `name_alias = "multi_ap_prod"`
`prio = "level1"`

```
resource "aci_application_epg" "prod_web" {
  [redacted] =
  application_profile.development_multi_app.id
  name = "web"
  name_alias = "Nginx"
  relationfv_rs_bd = [redacted]
```

Question 8

Question Type: MultipleChoice

```
resource "aci_filter" "db_traffic" {
  tenant_dn = [redacted]
  name = "db_traffic"
}
```

A DevOps engineer must build a Docker image to containerize an application. Then the image must be pushed to a repository on Docker Hub in a CI/CD pipeline using GitHub Actions.

Which approach securely encrypts the Docker Hub access token as an environment variable within the CI/CD pipeline?

```
resource "aci_filter_entry" "userdb" {
  filter_dn = [redacted]
  name = "userdb"
  [redacted] = "ip"
  protocol = "tcp"
  destination_from_port = "3306"
}
```

Options:

- A- Store the access token with GitHub environment variables
- B- Store the access token with GitHub encrypted secrets

C- Store the access token in an environment file in the repository

D- Hard code the access token in the repository with Base64 encoding

<code>aci_filter.db_traffic.id</code>	<code>fic.id</code>	<code>application_profile_dn</code>
<code>ether_t</code>	<code>tion_tenant.id</code>	<code>ether_t</code>
<code>aci_bridge_domain.production_bd.id</code>	<code>id</code>	<code>name</code>

Answer:

B

Question 9

Question Type: MultipleChoice

A development team uses Kubernetes for application development. Any changes on ConfigMap are performed manually for each development, test, and production environment. The edits are performed to deploy applications. This approach causes inconsistent deployments across all environments.

Which practice improves the consistency of the deployments?

Options:

- A-** Implement environment variables within the ConfigMaps and store the variable definitions separately from the master branch where the ConfigMaps are stored
- B-** Generate the ConfigMaps specific to the environment by using a templating language such as Jinja2 and store the ConfigMaps in unique branches of a repository
- C-** In the master branch where the ConfigMaps are stored, create a branch for each environment that contains an environment-specific ConfigMap.

D- Create a unique repository for each environment that contains ConfigMaps for that environment to ensure that each environment can be deployed independently

Answer:

A

Question 10

Question Type: DragDrop

Refer to the exhibit.

vices. To streamline the process, CO/CD



Refer to the exhibit.

Answer

Question 11

Question Type: MultipleChoice


```

import queue
import logging
from logging.handlers import QueueHandler, QueueListener

class Formatter:
    def __init__(self, formatters, default_formatter):
        self._formatters = formatters
        self._default_formatter = default_formatter
    def format(self, record):
        logger = logging.getLogger(record.name)
        while logger:
            if logger.name in self._formatters:
                formatter = self._formatters[logger.name]
                break
            else:
                logger = logger.parent
        else:
            formatter = self._default_formatter
        return formatter.format(record)

def main():
    que = queue.Queue(-1)
    queue_handler = QueueHandler(que)
    handler = logging.StreamHandler()
    listener = QueueListener(que, handler)
    root = logging.getLogger()
    root.addHandler(queue_handler)
    handler.setFormatter(Formatter({
        source: logging.Formatter('%(message)s -> ' + source),
        source + '.' + subsources: logging.Formatter('%(message)s -> ' + source +
        '.' + subsources),
    },
        logging.Formatter('%(message)s -> <default>'),
    ))
    listener.start()
if __name__ == "__main__":
    main()

```

A Python script implements a logger server. The log receives a message from Base that contains this text: TextMessage. How is the log formatted?

Options:

A- Base Alter: TextMessage

B- Undefined: TextMessageBase

C- TextMessage -> Base

D- TextMessage -> Alter Base

Answer:

B

Question 12

Question Type: DragDrop

-

A developer is creating an application where each service uses a different operating system. The application components need to be isolated but must have the ability to communicate with each other.

Drag and drop the entries from the left into the order on the right to create a Dockerfile that will accomplish this goal.

```
ENV CONFIG_PATH=/etc/application/conf/
```

step 1

```
ENTRYPOINT /path/to/the/app/entrypoint.sh
```

step 2

```
FROM example.com/application:latest
```

step 3

```
ADD config.ini ${CONFIG_PATH}
```

step 4

To Get Premium Files for 300-910 Visit

<https://www.p2pexams.com/products/300-910>

For More Free Questions Visit

<https://www.p2pexams.com/cisco/pdf/300-910>

