2022년 클라우드컴퓨팅 친경경기대회

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **직 종 명** | Cloud Computing | **과 제 명** | Web Service  Provisioning | **과제 번호** | 제1과제 |
| **경기시간** | 4시간 | **비 번 호** |  | **심사위원**  **확 인** | (인) |

|  |  |  |  |
| --- | --- | --- | --- |
| **번호** | **채점 기준** | 내용 | **배점** |
| 1 | 개발 환경 | 개발 용 PC 접근 | 1.3 |
| 2 | 웹 서비스 접근 및 가상 사설 네트워크  연결 | 웹 사이트 접근 여부 | 0.7 |
| 3 | 개발자 PC VPN 연결 | 1.5 |
| 4 | VPN 정상 구성 확인 | 1.5 |
| 5 | 내부 서비스 접근 | 1.5 |
| 6 | 내부 서비스 보안 | 1.5 |
| 7 | DNS 질의 | 1.5 |
| 8 | 제어 영역 | Control Host 접근 | 1.2 |
| 9 | DHCP Option 확인 | 0.6 |
| 10 | 컨테이너  오케스트레이션 | 쿠버네티스 클러스터 구축 | 1.5 |
| 11 | 쿠버네티스 노드 구축 | 1.5 |
| 12 | namespace | 0.2 |
| 13 | Deployment | 1.5 |
| 14 | Service | 1.5 |
| 15 | Node | 0.4 |
| 16 | Pods | 1.1 |
| 17 | 모니터링 | CloudWatch Logs Stream | 0.7 |
| 18 | Log Monitoring | 1.5 |
| 19 | 고 가용성 | Load Balancer | 1.1 |
| 20 | Load Balancer Target Group | 1.5 |
| 21 | 네트워크 | VPC | 1.5 |
| 22 | DHCP Option Set | 1.5 |
| 23 | Public Subnet | 1.5 |
| 24 | Auto Assign IP | 0.2 |
| 25 | Private Subnet | 1.5 |
| 26 | 인스턴스 | Instance Type | 1.5 |
| 27 | 개발자 | Developer DNS 질의 | 1.5 |
| 28 | 내부 API 서비스 접근 | 1.5 |
| 29 | 공인 IP | 1 |
| 30 | 도메인 네임 서비스 | Route 53 | 1.5 |
| 31 | Records | 1.5 |
| 32 | 가상 사설 네트워크 | Client VPN 구성확인 | 1.5 |
| 합 계 | | | 40 |

|  |  |
| --- | --- |
| **채점 번호** | **배점 항목** |
| **주의**  **사항** | **- 특별한 지시가 없는 한 모든 채점은 developer-online-client에서 진행합니다.**  **- 빨간 네모 박스가 되어 있는 그림은, 빨간 네모박스만 확인합니다.**  **- 채점이 종료되면, 선수는 추가 비용 소모 방지를 위해서 모든 리소스를 제거합니다.**  **- Region은 ap-northeast-2를 사용합니다.**  **- 직렬 콘솔 환경에서는 명령어가 이상하게 입력이 되는 현상이 있습니다.**  **오타 없이 선수가 직접 타이핑하여 채점을 진행합니다. (복사 붙여넣기는 사용 X)**  **- 암호가 필요로 하는 항목은 "Skill53##"을 입력합니다.** |
| **1** | 선수가 알아서 "developer-online-client" 라는 인스턴스에 ec2-user 사용자로 접근합니다.  접근 완료 후 아래와 같은 창이 떴는지 확인 |
| **2** | 아래의 명령어를 입력합니다.  $ curl --connect-timeout 5 -XGET <선수의 NLB DNS 주소>    위와 같이 Connection timed out 이라는 문구가 출력되었는지 확인 |
| **3** | 아래의 명령어를 입력합니다.  Firefox 실행 후 ec2로 이동합니다.      developer-online-client 라는 Name Tag를 가진 Instance를 선택 > Monitor and troubleshoot > EC2 Serial Console > Connect 클릭  그 후 아래와 같은 계정 정보로 로그인 진행    $ ID=`aws sts get-caller-identity --query Account --output text`  $ aws s3 cp s3://wsc-s3-bucket-$ID/vpn-config/client.ovpn ./  $ sudo amazon-linux-extras install epel -y ; sudo yum update -y  $ sudo yum install openvpn -y  $ sudo openvpn --config client.ovpn --daemon  $ sudo tail /var/log/messages  아래와 같이 "Initialization Sequence Completed"라는 메시지가 출력되었는 지 확인    ping 1.1.1.1 |
| **4** | 3번에서 접근한 직렬 콘솔에서 이어서 진행합니다.  $ ip address    위와 같이 tun0 이라는 Tunnel Interface가 생겼는지 확인하고 IP 대역이 172.29.x.x 인지 확인 |
| **5** | 3번 항목에서 접근한 직렬 콘솔에서 이어서 진행합니다.  $ curl -XPOST -d '{"nickname": "<심사위원이 원하는 문구 입력>", "email": "test01@wsc-france.internal"}' api.wsc-france.internal/users    위와 같이 명령어를 입력했을 때 {"nickname":"<심사위원이 입력한 문구>","email":"test01@wsc-france.internal"}가 출력됐는지 확인  $ curl -sS -XGET api.wsc-france.internal/users | jq '."test01@wsc-france.internal"'    // 해당 명령어는 Load Balancer 특성상 출력이 안될 수도 있으므로, 선수가 원하면 최대 8번까지 재실행 가능합니다.  위와 같이 nickname 항목에 위에서 심사위원이 입력한 문구와 email  항목에 "test01@wsc-france.internal"라는 문구가 쓰여 있는지 확인 |
| **6** | 3번에서 연결한 직렬콘솔에서 이어 진행합니다.  $ curl http://<선수의 ALB DNS주소>/users    위와 같이 "Error: Your not Admin" 이라는 문구가 출력되었는지 확인  $ wget http://<선수의 ALB DNS 주소>/users    ERROR 500이라는 문구가 출력되었는지 확인 |
| **7** | 3번에서 연결한 직렬콘솔에서 이어 진행합니다.  $ nslookup api.wsc-france.internal    위와 같이 질의에 대한 응답이 1개 이상 왔는지 확인  $ nslookup control.wsc-france.internal  만약 둘 중 1개라도 안될 경우 해당 항목 0점 처리합니다. |
| **8** | 3번에서 연결한 직렬콘솔에서 이어 진행합니다.  $ ssh ec2-user@control.wsc-france.internal -p30245    그 후 아래와 같은 창이 떴는지 확인 |
| **9** | 7번에서 연결한 환경 그대로 이어 진행합니다.  $ sudo yum install ntp -y  $ ntpstat | grep NTP ; ntptime | grep returns    위와 같이 출력되었는지 확인 및 NTP server의 주소가 169.254.169.123인지 확인  $ cat /etc/resolv.conf |
| **10** | 8번 항목에서 진행한 환경에서 이어서 진행합니다.  $ aws eks describe-cluster --name wscf-eks-cluster --region ap-northeast-2 --query 'cluster.{name:name, version:version}'    위와 같은 출력이 나왔는지 확인 |
| **11** | Firefox 실행 후 EKS 서비스로 이동      wscf-eks-cluster > Configuration > Compute 클릭 후 Node groups의 이름이 "wscf-worker-ng"인지 확인 후 EC2로 이동    위와 같이 wscf-worker-node 라는 인스턴스가 존재하는 지 확인 |
| **12** | 8번 항목에서 진행한 환경에서 이어서 진행합니다.  $ kubectl get ns | grep wscf-api    위와 같이 wscf-api라는 namespace가 존재하는 지 확인 |
| **13** | 8번 항목에서 진행한 환경에서 이어서 진행합니다.  $ kubectl get deployment -n wscf-api    위와 같이 wscf-deployment라는 Deployment가 존재하는지 확인 |
| **14** | 8번 항목에서 진행한 환경에서 이어서 진행합니다.  $ kubectl get svc -n wscf-api    wscf-service라는 Service가 존재하는지 확인 |
| **15** | $ kubectl get nodes    위와 같이 Node의 Name에 10.105.200.x과 10.105.202.x 가 포함되어 있는지 확인 |
| **16** | $ kubectl get pods -n wscf-api    위와 같이 2대 이상의 Pod가 존재하고, 모두 Running 상태인지 확인 |
| **17** | Firefox 실행 후 CloudWatch로 이동        Log groups > /aws/wsc/France로 이동 후 wscf-deployment가 존재하는지 확인 |
| **18** | developer-online-client 인스턴스에서 채점 진행합니다. (VPN 연결이 되어있는 상태에서 진행)  $ curl api.wsc-france.internal/users  위 명령어를 10번 입력합니다. 그 후 CloudWatch로 돌아와 wscf-deployment stream에 아래와 같은 Log가 10번 정도 출력 되었는지 확인합니다. |
| **19** | Firefox 실행 후 EC2로 이동합니다.    Load Balancers 로 이동  그리고 아래와 같이 총 두 개의 Load Balancer가 존재하고 Type이 wscf-back-alb는 application 이고, wscf-front-nlb는 network인지 확인    그리고 각 로드 밸런서를 클릭해서 Scheme가 internal 인지 확인 |
| **20** | wscf-front-nlb 로그 밸런서 클릭 > Listener > wscf-alb-tg  아래와 같이 Target Type이 Application Load Balancer이고 Load Balancer가 wscf-front-nlb인지 확인해준다.    그리고 해당 Target Group의 Health가 Healthy인지 확인 |
| **21** | VPC로 이동    위와 같이 wscf-manage.vpc의 IPv4 CIDR가 10.105.0.0/16인지 확인 |
| **22** | VPC 이동 후 DHCP Options Sets으로 이동    그 후 wsc-dhcp-ops의 Details 항목에서 아래와 같이 설정 되어 있는지 확인    만약 이중 1개라도 없을 경우 해당 항목 0점 처리합니다. |
| **23** | Subnet으로 이동합니다.      위와 같이 wscf-manage.pub-a의 IPv4 CIDR이 10.105.100.0/24 이면서 Availability Zone이 ap-northeast-2a 이고 wscf-manage.pub-c의 IPv4 CIDR이 10.105.102.0/24 이면서 Availability Zone이 ap-northeast-2c 인지 확인 |
| **24** | 해당 Public Subnet들이 각가 아래와 같이 설정이 되어있는지 확인 |
| **25** | Subnet으로 이동합니다.    위와 같이 wscf-manage.priv-a의 IPv4 CIDR이 10.105.200.0/24 이면서 Availability Zone이 ap-northeast-2a 이고 wscf-manage.priv-b의 IPv4 CIDR이 10.105.202.0/24 이면서 Availability Zone이 ap-northeast-2b인지 확인 |
| **26** | EC2로 이동 후 진행합니다.    위와 같이 검색 후 아래와 같이 wscf-worker-node와, wscf-manage-control-host 인스턴스가 존재하는 지 확인 |
| **27** | 우선 developer-online-clinet에 접근 후 진행합니다.  $ sudo killall openvpn  $ nslookup api.wsc-france.internal    위와 같이 Query가 안되는지 확인 |
| **28** | 27에서 진행한 환경에서 진행합니다.  $ curl api.wsc-france.internal/health    위와 같이 응답이 안오는지 확인 |
| **29** | 27번에서 진행한 환경에서 진행합니다.  $ curl ifconfig.io      위에서 출력된 IP가 EIP에 존재하는지 확인 |
| **30** | Route53으로 이동      Hosted zones > wsc-france.internal > Hosted zone details    위와 같이 Type이 Private hosted zone이고 Associated VPCs의 VPC Id가 wscf-manage.vpc의 ID와 같은지 확인  만약 VPC ID가 다르거나 1개 이상 존재할 경우 해당 항목 0점처리 |
| **31** | 위와 같이 api.wsc-frace.internal과 control.wsc-france.internal Record가 있는지 확인 하고 api.wsc-frace.internal를 클릭해서 Alias가 Yes인지 확인 |
| **32** | developer-online-client 인스턴스에 VPN 연결을 아래와 같은 명령어를  입력하여 해제 후 해당 인스턴스에 SSH 접근 후 진행합니다.  $ sudo killall openvpn  developer-online-client에서 진행  $ aws ec2 describe-client-vpn-endpoints --region ap-northeast-2          위와 같은 내용이 포함되어 있는지 확인 |