

프로젝트 최종 보고서

과제 현황	
수행기간	2023.01.16 ~ 2023.01.27
프로젝트 명	효율적인 네트워크 구축을 위한 웹 서버 자동화
팀명	지금 2순간
팀원	백상목 안방호 유지현

보고서	
작품명	온프레미스 구축 script 활용한 자동화
# key words	#Kubernetes #Ansible #shell #Wordpress #mysql #helm
1. 프로젝트 개요	기존 온프레미스 환경 구성 중 반복되는 작업들로 인해 휴먼에러가 발생하고, 작업속도도 현저히 낮아집니다. 이를 스크립트화 시켜, 속도를 증진시키고 에러를 방지해 자동화를 구축했습니다.
2. 프로젝트 구성도	<p>Namespace</p> <ul style="list-style-type: none">svc1svc2svc3
3. 프로젝트 내용	<p>1. 상황</p> <ul style="list-style-type: none">- 같은 네트워크 내의 통신이 되는 PC 5대 <p>2. 목적</p> <ul style="list-style-type: none">- Ansible master server 1, kubernetes master server 1, kubernetes node 3- Wordpress, mysql 컨테이너 배포 및 연결

2. 자동화 방식

- Script에 변수 입력(각 PC IP)
- Script를 이용해 Ansible 배포
- Script를 이용해 Ansible로 아래 명령 실행

- 각 kubernetes 서버에 Script 가동 후 kubernetes 설치
- Wordpress.yaml, Mysql.yaml 파일 kubernetes master server에 전송
- 컨테이너 실행 및 연결

1. Linux Shell 활용 Ansible 설치 및 hostname 변경 & Key 배포

```
# ansible_install.txt
1 #!/bin/bash
2 #키 만들고 푸르기
3
4
5 #변수 생성
6 var00="k8sa" # ansible
7 var11="k8sb" # master
8 var22="k8sn1" #node1
9 var33="k8sn2" #node2
10 var44="k8sn3" #node3
11 var1="10.0.0.101" #master
12 var2="10.0.0.102" #node1
13 var3="10.0.0.103" #node2
14 var4="10.0.0.104" #node3
15
16 mkdir ansible_install
17 cd ./ansible_install
18
19 # 이 큰 바꾸기 #etc/hostname
20 cat > /etc/hostname << EOF
21 ${var00}.labs.local
22 EOF
23
24 #유지생성
25 useradd admin
26 echo 'Iti' | passwd --stdin admin
27
28
29 #expect 설치
30 yum install -y epel-release expect
31
32 #ssh 키 생성 파일 생성
33 cat > ./ssh.exp << EOF
34 #!/usr/bin/expect
35 spawn sudo -u admin ssh-keygen -t rsa
36 expect {
37     "(/home/admin/.ssh/id_rsa):" { send "\n"; exp_continue }
38 }
39 expect {
40     "(empty for no passphrase):" { send "\n"; exp_continue }
41 }
42 expect {
43     "passphrase again:" { send "\n"; exp_continue }
44 }
45 expect eof
46 EOF
47
48 #모드 변경
49 chmod 777 ./ssh.exp
50
51 #키 생성 실행
52 ./ssh.exp
53
54 #expect 파일 생성
55 cat > ./aa.exp << EOF
56 #!/usr/bin/expect
57 set username [lindex $argv 0]
58 set password [lindex $argv 1]
59 set hostname [lindex $argv 2]
60 if {[llength $argv] == 0} {
61     send user "Usage: scriptname username \'password\' hostname \n"
62     exit 1
63 }
64 spawn ssh -o StrictHostKeyChecking=no \${username}@${hostname}
65 expect {
66     "(yes/no)" { send "yes\n"; exp_continue }
67     "password:" { send "\${password}\n"; exp_continue }
68     "Last login:"
69     "useradd admin\n"
70     "send \"passwd admin\n"
71     "새 암호:" { send "\${password}\n"; exp_continue }
72     "새 암호 재입력:" { send "\${password}\n"; exp_continue }
73     send user "nm SSH is success \n"
74     EOF
75     chmod 777 ./aa.exp
76 }
77
78 #필수 확인 유지생성
79 ./aa.exp root Iti ${var1}
80 ./aa.exp root Iti ${var2}
81 ./aa.exp root Iti ${var3}
82 ./aa.exp root Iti ${var4}
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

4. 내용

```
# ansible_install.txt
144 #ansible 설치
145 yum install -y ansible
146 yum install -y python3
147
148 #etc/ansible/hosts
149 cat > /etc/ansible/hosts << EOF
150 ${var1}
151 ${var2}
152 ${var3}
153 ${var4}
154 [master]
155 ${var1}
156 [node]
157 ${var2}
158 ${var3}
159 ${var4}
160 EOF
161
162 #ssh 키 생성 파일 생성
163 cat > ./ssh1.exp << EOF
164 #!/usr/bin/expect
165 set username [lindex $argv 0]
166 set password [lindex $argv 1]
167 set hostname [lindex $argv 2]
168 if {[llength $argv] == 0} {
169     send user "Usage: scriptname username \'password\' hostname \n"
170     exit 1
171 }
172 spawn sudo -u admin ssh admin@${username}
173 expect {
174     "(yes/no)" { send "yes\n"; exp_continue }
175 }
176 send user "nm SSH is success \n"
177 send "exit \n"
178 expect eof
179 EOF
180
181 #모드 변경
182 chmod 777 ./ssh1.exp
183
184 ./ssh1.exp ${var1} a a
185 ./ssh1.exp ${var2} a a
186 ./ssh1.exp ${var3} a a
187 ./ssh1.exp ${var4} a a
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

2. Ansible을 통한 Shell script 실행(K8S Master & Node Server 생성)

	<div> <pre> 1 #!/bin/bash 2 3 #반드시 master.sh 로 작업!!***** 4 #쿠버네티스 마스터 서버 5 var1="10.0.0.101" #master 6 7 8 # selinux off 9 setenforce 0 10 11 sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/sysconfig/selinux 12 13 # firewall off & disable 14 systemctl stop firewalld 15 systemctl disable firewalld 16 # Master 17 # firewall-cmd --add-port={80,443,6443,2379,2380,10250,10251,10252,30000-32767} --permanent 18 # Node 19 # firewall-cmd --add-port={80,443,10250,30000-32767}/tcp --permanent 20 21 # swap off 22 swapoff -a 23 sed -i 's/swap/s/^(.*)\$/\1/g' /etc/fstab 24 25 # overlay & iptables Module load 26 cat > /etc/modules-load.d/k8s.conf << EOF 27 overlay 28 br_netfilter 29 EOF 30 31 modprobe overlay 32 modprobe br_netfilter 33 34 # iptables & NAT 설정 35 cat > /etc/sysctl.d/k8s.conf << EOF 36 net.bridge.bridge-nf-call-iptables = 1 37 net.bridge.bridge-nf-call-iptables = 1 38 net.ipv4.ip_forward = 1 39 EOF 40 sysctl --system 41 42 # K8S Repository 쿠버네티스 리포지토리 43 cat > /etc/yum.repos.d/k8s.repo << EOF 44 [kubernetes] 45 name=kubernetes 46 baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-\$basearch 47 enabled=1 48 gpgcheck=1 49 repo_gpgcheck=0 50 gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages. </pre> </div> <div> <pre> 48 exclude=kubelet kubeadm kubectl 49 EOF 50 51 # kubeadm, kubelet, kubectl Package Install # 쿠버네티스 설치 52 yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes 53 54 systemctl enable kubelet 55 systemctl start kubelet 56 57 # container runtime install # 컨테이너 설치 58 yum install -y yum-utils 59 yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo 60 yum install -y containerd.io 61 mkdir -p /etc/containerd 62 containerd config default sudo tee /etc/containerd/config.toml 63 sed -i'' -r -e "/containerd.default_runtime.options/a\SystemdCgroup=true" /etc/containerd/config 64 systemctl enable containerd 65 systemctl restart containerd 66 67 #### master*** 68 69 kubeadm config print init-defaults > kubeadm-init.yaml 70 cat >> kubeadm-init.yaml << EOF 71 --- 72 apiVersion: kubelet.config.k8s.io/v1beta1 73 kind: KubeletConfiguration 74 cgroupDriver: systemd 75 EOF 76 sed -i 's/name: node/name: master/g' kubeadm-init.yaml 77 sed -i 's/advertiseAddress: 1.2.3.4/advertiseAddress: "\${var1}"/' kubeadm-init.yaml 78 kubeadm init --config=kubeadm-init.yaml 79 80 mkdir -p \$HOME/.kube 81 cp -i /etc/kubernetes/admin.conf \$HOME/.kube/config 82 chown \$(id -u):\$(id -g) \$HOME/.kube/config 83 export KUBECONFIG=/etc/kubernetes/admin.conf 84 85 yum install -y wget 86 # https://docs.tigera.io/calico/latest/getting-started/kubernetes/self-managed-onpremise 87 curl https://raw.githubusercontent.com/projectcalico/calico/v3.25.1/manifests/calico.yaml -O 88 kubectl apply -f calico.yaml </pre> </div> <div> <pre> 1 #!/bin/bash 2 3 #반드시 node.sh 로 작업!!***** 4 #쿠버네티스 노드 / 마스터 서버 5 6 yum install -y rdate 7 rdate -s time.bora.net 8 9 # selinux off 10 setenforce 0 11 sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/sysconfig/selinux 12 13 # firewall off & disable 14 systemctl stop firewalld 15 systemctl disable firewalld 16 # Master 17 # firewall-cmd --add-port={80,443,6443,2379,2380,10250,10251,10252,30000-32767} --permanent 18 # Node 19 # firewall-cmd --add-port={80,443,10250,30000-32767}/tcp --permanent 20 21 # swap off 22 swapoff -a 23 sed -i 's/swap/s/^(.*)\$/\1/g' /etc/fstab 24 25 # overlay & iptables Module load 26 cat > /etc/modules-load.d/k8s.conf << EOF 27 overlay 28 br_netfilter 29 EOF 30 31 modprobe overlay 32 modprobe br_netfilter 33 34 # iptables & NAT 설정 35 cat > /etc/sysctl.d/k8s.conf << EOF 36 net.bridge.bridge-nf-call-iptables = 1 37 net.bridge.bridge-nf-call-iptables = 1 38 net.ipv4.ip_forward = 1 39 EOF 40 sysctl --system 41 42 # K8S Repository 쿠버네티스 리포지토리 43 cat > /etc/yum.repos.d/k8s.repo << EOF 44 [kubernetes] 45 name=kubernetes 46 baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-\$basearch 47 enabled=1 </pre> </div> <div> <pre> 48 gpgcheck=1 49 repo_gpgcheck=0 50 gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.g 51 exclude=kubelet kubeadm kubectl 52 EOF 53 54 # kubeadm, kubelet, kubectl Package Install # 쿠버네티스 설치 55 yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes 56 57 systemctl enable kubelet 58 systemctl start kubelet 59 60 # container runtime install # 컨테이너 설치 61 yum install -y yum-utils 62 yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo 63 yum install -y containerd.io 64 mkdir -p /etc/containerd 65 containerd config default sudo tee /etc/containerd/config.toml 66 sed -i'' -r -e "/containerd.default_runtime.options/a\SystemdCgroup=true" /etc/contai 67 systemctl enable containerd 68 systemctl restart containerd </pre> </div>
<div>5. 첨부파일</div>	<div> <ul style="list-style-type: none"> - ansible_install.txt - master.txt - node.txt - start.txt </div>