Element Controller
Installation Manual

Version 0.9

2017年10月

更新履歴

日付	版数	記入者	変更箇所	変更内容
2017/10	0.9	−	−	初版登録

目次

表目次

1. はじめに

1.1. 目的

本ドキュメントは、Element Controller （以下 EC）に含まれる EC メインモジュールのインストールマニュアルです。本ソフトウェアを動作するにあたり、事前に内容をご確認のうえ、ご使用下さい。

1.2. 適用範囲

本ドキュメントの適用範囲は、EC メインモジュールの構成品になります。

上述以外に、本書を適用することは出来ません。

1.3. 文言表現の意味

本文では、以下のような文字表現をしています。内容をよく理解してから、本文をお読み下さい。

- **<実行サーバ：XXX>**：太字山括弧囲み。
  この場合は、コマンドを実行するサーバ XXX を意味します。
- **【XX XX】**： 太字中括弧囲み。
  この場合は、Linux へのコマンド投入を意味します。
- **X [Enter]**：太字＋[Enter]。
  この場合は、コンソール画面での、キー入力＋エンターキー押下を意味します。

1.4. 商標

本ドキュメントに記載されている会社名、商品名は、各社の登録商標または商標です。

Linux®:

Linus Torvalds 氏の米国及びその他の国における登録商標あるいは商標です。

PostgreSQL@:

PostgreSQL の米国およびその他の国における商標です。

1.5. 付属品の構成

本ドキュメントのインスール手順に必要な付属品について以下の表 1-1 付属品に示す。
また、備考に事前ダウンロードの記述がある項目については本資料のインストールを実施する前にダウンロードを行い入手すること。

表 1-1 付属品

No	フォルダ構成	ファイル名	説明	備考
1.	ec_main	-	-	-
2.	bin	boot.sh	装置間 IF 制御機能部装置起動通知スクリプト	
3.		ec_ctl.sh	EC 起動スクリプト	
4.		linkdown.sh	SNMPTrap 機能部リンクダウン通知スクリプト	
5.		linkup.sh	SNMPTrap 機能部リンクアップ通知スクリプト	
6.	lib	EcMainModule.jar	EC メインモジュール	
7.		NetConf.jar	NetConf ライブラリ	
8.		antlr-2.7.7.jar	使用ライブラリ	事前ダウンロード
9.		c3p0-0.9.2.1.jar	使用ライブラリ	事前ダウンロード
10.		commons-io-2.5.jar	使用ライブラリ	事前ダウンロード
11.		dom4j-1.6.1.jar	使用ライブラリ	事前ダウンロード
12.		ganymed-ssh2-build210.jar	使用ライブラリ	事前ダウンロード
13.		geronimo-jta_1.1_spec-1.1.1.jar	使用ライブラリ	事前ダウンロード
14.		gson-2.7.jar	使用ライブラリ	事前ダウンロード
15.		hibernate-c3p0-5.0.10.Final.jar	使用ライブラリ	事前ダウンロード
16.		hibernate-commons-annotations-5.0.1.Final.jar	使用ライブラリ	事前ダウンロード
17.		hibernate-core-5.0.10.Final.jar	使用ライブラリ	事前ダウンロード
18.		hibernate-jpa-2.1-api-1.0.0.Final.jar	使用ライブラリ	事前ダウンロード
19.		hk2-api-2.5.0-b05.jar	使用ライブラリ	事前ダウンロード
20.		hk2-locator-2.5.0-b05.jar	使用ライブラリ	事前ダウンロード
21.		hk2-utils-2.5.0-b05.jar	使用ライブラリ	事前ダウンロード
22.		javassist-3.18.1-GA.jar	使用ライブラリ	事前ダウンロード
23.		javax.annotation-api-1.2.jar	使用ライブラリ	事前ダウンロード
24.		javax.inject-2.5.0-b05.jar	使用ライブラリ	事前ダウンロード
25.		javax.ws.rs-api-2.0.1.jar	使用ライブラリ	事前ダウンロード
26.		jboss-logging-3.3.0.Final.jar	使用ライブラリ	事前ダウンロード
27.		jersey-client.jar	使用ライブラリ	事前ダウンロード
28.		jersey-common.jar	使用ライブラリ	事前ダウンロード
29.		jersey-container-servlet-core.jar	使用ライブラリ	事前ダウンロード
30.		jersey-guava-2.23.2.jar	使用ライブラリ	事前ダウンロード
31.		jersey-server.jar	使用ライブラリ	事前ダウンロード
32.		jetty-http-9.3.11.v20160721.jar	使用ライブラリ	事前ダウンロード
33.		jetty-io-9.3.11.v20160721.jar	使用ライブラリ	事前ダウンロード
34.		jetty-security-9.3.11.v20160721.jar	使用ライブラリ	事前ダウンロード
35.		jetty-server-9.3.11.v20160721.jar	使用ライブラリ	事前ダウンロード
36.		jetty-servlet-9.3.11.v20160721.jar	使用ライブラリ	事前ダウンロード
37.		jetty-util-9.3.11.v20160721.jar	使用ライブラリ	事前ダウンロード
38.		jsch-0.1.53.jar	使用ライブラリ	事前ダウンロード
39.		log4j-api-2.6.2.jar	使用ライブラリ	事前ダウンロード
40.		log4j-core-2.6.2.jar	使用ライブラリ	事前ダウンロード

41.			log4j-slf4j-impl-2.6.2.jar	使用ライブラリ	事前ダウンロード
42.			mchange-commons-java-0.2.3.4.jar	使用ライブラリ	事前ダウンロード
43.			org.eclipse.persistence.core.jar	使用ライブラリ	事前ダウンロード
44.			postgresql-9.4.1209.jre7.jar	使用ライブラリ	事前ダウンロード
45.			quartz-2.2.3.jar	使用ライブラリ	事前ダウンロード
46.			servlet-api-3.1.jar	使用ライブラリ	事前ダウンロード
47.			slf4j-api-1.7.21.jar	使用ライブラリ	事前ダウンロード
48.			slf4j-simple-1.7.21.jar	使用ライブラリ	事前ダウンロード
49.			snmp4j-2.5.0.jar	使用ライブラリ	事前ダウンロード
50.			validation-api-1.1.0.Final.jar	使用ライブラリ	事前ダウンロード
51.	conf		ec_main.conf	EC メインモジュール設定ファイル	
52.			hibernate.cfg.xml	Hibernate 設定ファイル	
53.			log4j2.xml	log4j2 設定ファイル	
54.	installer		-	-	-
55.		dhcp.v4.2.5	dhcp-4.2.5-42.el7.centos.x86_64.rpm	DHCP インストールパッケージ	事前ダウンロード
56.		ntp.v4.2	autogen-libopts-5.18-5.el7.x86_64.rpm	NTP インストールパッケージ	事前ダウンロード
57.			ntpdate-4.2.6p5-22.el7.centos.x86_64.rpm	NTP インストールパッケージ	事前ダウンロード
58.			ntp-4.2.6p5-22.el7.centos.x86_64.rpm	NTP インストールパッケージ	事前ダウンロード
59.		postgresql.v9.3.13	postgresql93-9.3.13-1PGDG.rhel7.x86_64.rpm	PostgreSQL インストールパッケージ	事前ダウンロード
60.			postgresql93-contrib-9.3.13-1PGDG.rhel7.x86_64.rpm	PostgreSQL インストールパッケージ	事前ダウンロード
61.			postgresql93-devel-9.3.13-1PGDG.rhel7.x86_64.rpm	PostgreSQL インストールパッケージ	事前ダウンロード
62.			postgresql93-libs-9.3.13-1PGDG.rhel7.x86_64.rpm	PostgreSQL インストールパッケージ	事前ダウンロード
63.			postgresql93-server-9.3.13-1PGDG.rhel7.x86_64.rpm	PostgreSQL インストールパッケージ	事前ダウンロード
64.			uuid-1.6.2-26.el7.x86_64.rpm	PostgreSQL インストールパッケージ	事前ダウンロード
65.			libxslt-1.1.28-5.el7.x86_64.rpm	PostgreSQL インストールパッケージ	事前ダウンロード
66.		snmptrapd.v5.7.2	net-snmp-5.7.2-24.el7.x86_64.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
67.			perl-HTTP-Tiny-0.033-3.el7.noarch.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
68.			perl-Pod-Perldoc-3.20-4.el7.noarch.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
69.			perl-podlators-2.5.1-3.el7.noarch.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
70.			perl-Encode-2.51-7.el7.x86_64.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
71.			perl-Text-ParseWords-3.29-4.el7.noarch.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
72.			perl-Pod-Usage-1.63-3.el7.noarch.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
73.			perl-constant-1.27-2.el7.noarch.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
74.			perl-Time-Local-1.2300-2.el7.noarch.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
75.			perl-Storable-2.45-3.el7.x86_64.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
76.			perl-Socket-2.010-3.el7.x86_64.rpm	Net-SNMP インストールパッケージ	事前ダウンロード

77.				perl-Scalar-List-Utils-1.27-248.el7.x86_64.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
78.				perl-File-Temp-0.23.01-3.el7.noarch.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
79.				perl-Getopt-Long-2.40-2.el7.noarch.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
80.				perl-File-Path-2.09-2.el7.noarch.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
81.				perl-Exporter-5.68-3.el7.noarch.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
82.				perl-Carp-1.26-244.el7.noarch.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
83.				perl-PathTools-3.40-5.el7.x86_64.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
84.				perl-Pod-Escapes-1.04-286.el7.noarch.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
85.				perl-macros-5.16.3-286.el7.x86_64.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
86.				perl-threads-shared-1.43-6.el7.x86_64.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
87.				perl-threads-1.87-4.el7.x86_64.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
88.				perl-Time-HiRes-1.9725-3.el7.x86_64.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
89.				perl-Pod-Simple-3.28-4.el7.noarch.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
90.				perl-Filter-1.49-3.el7.x86_64.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
91.				perl-parent-0.225-244.el7.noarch.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
92.				net-snmp-agent-libs-5.7.2-24.el7_2.1.x86_64.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
93.				net-snmp-libs-5.7.2-24.el7_2.1.x86_64.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
94.				perl-5.16.3-286.el7.x86_64.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
95.				perl-libs-5.16.3-286.el7.x86_64.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
96.				perl-Data-Dumper-2.145-3.el7.x86_64.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
97.				net-snmp-5.7.2-24.el7_2.1.x86_64.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
98.				openssl098e-0.9.8e-29.el7.centos.3.x86_64.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
99.				glibc-2.17-105.el7.x86_64.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
100.				glibc-common-2.17-105.el7.x86_64.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
101.				lm_sensors-libs-3.3.4-11.el7.x86_64.rpm	Net-SNMP インストールパッケージ	事前ダウンロード
102.			java.v8u92	jdk-8u92-linux-x64.rpm	Java インストールパッケージ	事前ダウンロード
103.			pacemaker.v1.1.14-1.1	pacemaker-1.1.14-1.el7.x86_64.rpm	Pacemaker インストールパッケージ	事前ダウンロード
104.				corosync-2.3.5-1.el7.x86_64.rpm	Corosync インストールパッケージ	事前ダウンロード
105.				crmsh-2.1.5-1.el7.x86_64.rpm	crm コマンドインストールパッケージ	事前ダウンロード
106.				pcs-0.9.143-15.el7.x86_64.rpm	pcs コマンドインストールパッケージ	最新パッケージは 0.9.149-1 だが、試用版のため一つ前

					の ver を使用
107.			cluster-glue-1.0.12-2.el7.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
108.			cluster-glue-libs-1.0.12-2.el7.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
109.			corosynclib-2.3.5-1.el7.x86_64.rpm	Corosync 依存パッケージ	事前ダウンロード
110.			ipmitool-1.8.13-9.el7_2.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
111.			libqb-1.0-1.el7.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
112.			libtool-ltdl-2.4.2-21.el7_2.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
113.			libxslt-1.1.28-5.el7.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
114.			libyaml-0.1.4-11.el7_0.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
115.			lm_sensors-libs-3.3.4-11.el7.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
116.			nano-2.3.1-10.el7.x86_64.rpm	crm 依存パッケージ	事前ダウンロード
117.			net-snmp-agent-libs-5.7.2-24.el7_2.1.x86_64.rpm	Corosync 依存パッケージ	事前ダウンロード
118.			net-snmp-libs-5.7.2-24.el7_2.1.x86_64.rpm	Corosync 依存パッケージ	事前ダウンロード
119.			openhpi-libs-3.4.0-2.el7.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
120.			OpenIPMI-libs-2.0.19-11.el7.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
121.			OpenIPMI-modalias-2.0.19-11.el7.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
122.			pacemaker-cli-1.1.14-1.el7.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
123.			pacemaker-cluster-libs-1.1.14-1.el7.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
124.			pacemaker-libs-1.1.14-1.el7.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
125.			pacemaker-all-1.1.14-1.1.el7.noarch.rpm	Pacemaker 依存パッケージ	事前ダウンロード
126.			perl-5.16.3-286.el7.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
127.			perl-Carp-1.26-244.el7.noarch.rpm	Pacemaker 依存パッケージ	事前ダウンロード
128.			perl-constant-1.27-2.el7.noarch.rpm	Pacemaker 依存パッケージ	事前ダウンロード
129.			perl-Encode-2.51-7.el7.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
130.			perl-Exporter-5.68-3.el7.noarch.rpm	Pacemaker 依存パッケージ	事前ダウンロード
131.			perl-File-Path-2.09-2.el7.noarch.rpm	Pacemaker 依存パッケージ	事前ダウンロード
132.			perl-File-Temp-0.23.01-3.el7.noarch.rpm	Pacemaker 依存パッケージ	事前ダウンロード
133.			perl-Filter-1.49-3.el7.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
134.			perl-Getopt-Long-2.40-2.el7.noarch.rpm	Pacemaker 依存パッケージ	事前ダウンロード
135.			perl-HTTP-Tiny-0.033-3.el7.noarch.rpm	Pacemaker 依存パッケージ	事前ダウンロード
136.			perl-libs-5.16.3-286.el7.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
137.			perl-macros-5.16.3-286.el7.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
138.			perl-parent-0.225-244.el7.noarch.rpm	Pacemaker 依存パッケージ	事前ダウンロード
139.			perl-PathTools-3.40-5.el7.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
140.			perl-Pod-Escapes-1.04-286.el7.noarch.rpm	Pacemaker 依存パッケージ	事前ダウンロード
141.			perl-podlators-2.5.1-3.el7.noarch.rpm	Pacemaker 依存パッケージ	事前ダウンロード
142.			perl-Pod-Perldoc-3.20-4.el7.noarch.rpm	Pacemaker 依存パッケージ	事前ダウンロード
143.			perl-Pod-Simple-3.28-4.el7.noarch.rpm	Pacemaker 依存パッケージ	事前ダウンロード
144.			perl-Pod-Usage-1.63-3.el7.noarch.rpm	Pacemaker 依存パッケージ	事前ダウンロード

145.				perl-Scalar-List-Utils-1.27-248.el7.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
146.				perl-Socket-2.010-3.el7.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
147.				perl-Storable-2.45-3.el7.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
148.				perl-Text-ParseWords-3.29-4.el7.noarch.rpm	Pacemaker 依存パッケージ	事前ダウンロード
149.				perl-threads-1.87-4.el7.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
150.				perl-threads-shared-1.43-6.el7.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
151.				perl-TimeDate-2.30-2.el7.noarch.rpm	Pacemaker 依存パッケージ	事前ダウンロード
152.				perl-Time-HiRes-1.9725-3.el7.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
153.				perl-Time-Local-1.2300-2.el7.noarch.rpm	Pacemaker 依存パッケージ	事前ダウンロード
154.				pm_crmgen-2.1-1.el7.noarch.rpm	Pacemaker 依存パッケージ	事前ダウンロード
155.				pm_diskd-2.2-1.el7.x86_64.rpm	Diskd RA パッケージ	事前ダウンロード
156.				pm_extras-2.2-1.el7.x86_64.rpm	VIPCheck RA パッケージ	事前ダウンロード
157.				pm_logconv-cs-2.2-1.el7.noarch.rpm	Pacemaker 依存パッケージ	事前ダウンロード
158.				psmisc-22.20-9.el7.x86_64.rpm	Pacemaker 依存パッケージ	事前ダウンロード
159.				pssh-2.3.1-5.el7.noarch.rpm	crm 依存パッケージ	事前ダウンロード
160.				python-clufter-0.50.4-1.el7.x86_64.rpm	pcs 依存パッケージ	事前ダウンロード
161.				python-dateutil-1.5-7.el7.noarch.rpm	pcs 依存パッケージ	事前ダウンロード
162.				python-lxml-3.2.1-4.el7.x86_64.rpm	pcs 依存パッケージ	事前ダウンロード
163.				resource-agents-3.9.7-1.2.6f56.el7.x86_64.rpm	仮想 IPRA を含む標準 RA パッケージ	事前ダウンロード
164.				ruby-2.0.0.598-25.el7_1.x86_64.rpm	pcs 依存パッケージ	事前ダウンロード
165.				rubygem-bigdecimal-1.2.0-25.el7_1.x86_64.rpm	pcs 依存パッケージ	事前ダウンロード
166.				rubygem-io-console-0.4.2-25.el7_1.x86_64.rpm	pcs 依存パッケージ	事前ダウンロード
167.				rubygem-json-1.7.7-25.el7_1.x86_64.rpm	pcs 依存パッケージ	事前ダウンロード
168.				rubygem-psych-2.0.0-25.el7_1.x86_64.rpm	pcs 依存パッケージ	事前ダウンロード
169.				rubygem-rdoc-4.0.0-25.el7_1.noarch.rpm	pcs 依存パッケージ	事前ダウンロード
170.				rubygems-2.0.14-25.el7_1.noarch.rpm	pcs 依存パッケージ	事前ダウンロード
171.				ruby-irb-2.0.0.598-25.el7_1.noarch.rpm	pcs 依存パッケージ	事前ダウンロード
172.				ruby-libs-2.0.0.598-25.el7_1.x86_64.rpm	pcs 依存パッケージ	事前ダウンロード
173.				pacemaker_install.sh	パッケージインストーラ	
174.			httpd.v2.4.6	apr-1.4.8-3.el7.x86_64.rpm	httpd インストールパッケージ	事前ダウンロード
175.				apr-util-1.5.2-6.el7.x86_64.rpm	httpd インストールパッケージ	事前ダウンロード
176.				mailcap-2.1.41-2.el7.noarch.rpm	httpd インストールパッケージ	事前ダウンロード
177.				httpd-tools-2.4.6-40.el7.centos.4.x86_64.rpm	httpd インストールパッケージ	事前ダウンロード
178.				httpd-2.4.6-40.el7.centos.4.x86_64.rpm	httpd インストールパッケージ	事前ダウンロード
179.				httpd-manual-2.4.6-40.el7.centos.4.noarch.rpm	httpd インストールパッケージ	事前ダウンロード
180.				mod_ssl-2.4.6-40.el7.centos.4.x86_64.rpm	httpd インストールパッケージ	事前ダウンロード
181.			tftp.v5.2	xinetd-2.3.15-12.el7.x86_64.rpm	tftpd インストールパッケージ	事前ダウンロード
182.				tftp-5.2-12.el7.x86_64.rpm	tftpd インストールパッケージ	事前ダウンロード
183.				tftp-server-5.2-12.el7.x86_64.rpm	tftpd インストールパッケージ	事前ダウンロード
184.		database		-	-	-

185.			create_table.sql	テーブル作成スクリプト	
		RA	-	-	-
186.			ra_config.csv	リソースエージェント設定ファイル	
187.			ra_config.xlsx	リソースエージェント設定ファイル	
188.			ec	リソースエージェント	
189.			snmptrapd	リソースエージェント	

2. 動作環境

2.1. EC メインモジュール起動サーバ

以下の Linux マシン環境上での動作を推奨します。

表 2-1 推奨ハードウェア構成

No.	機器	動作保証
1.	OS	CentOS7.2 x86_64
2.	CPU	Intel(R) Xeon(R) CPU E5-2420 v2 @2.20GHz 6Core/12Thread 以上
3.	メモリ	32GB 以上
4.	ハードディスク空き容量	空き容量 500G 以上
5.	NIC	2 ポート以上

3. コントローラサーバインストール準備

3.1. OS インストール

OS インストール方法は「Fabric_Controller_Installation_Manual」を参照のこと。

4. コントローラサーバインストール

本章の作業は、特定のユーザ指定がない限りはルートユーザで実行すること。

＜実行ホスト：ACT/SBY/DB＞

インストール時にファイルを配置するワーク用のフォルダ（以下、ワークフォルダ）を作成する。
（コントローラサーバインストールが完了した時点で削除する）

【mkdir ~/setup】[Enter]

作成後、1.5 付属品の構成の ec_main フォルダをワークフォルダに配置する。

4.1. 環境インストール

4.1.1. ファイアウォール設定

4.1.1.1. ファイアウォールの確認

＜実行ホスト：ACT/SBY/DB＞

ファイアウォールが設定されているかの確認を行う。

【firewall-cmd --state】[Enter]

ファイアウォールが設定されている場合、画面には以下のメッセージが表示される。

running

この場合続けて、4.1.1.2～4.1.1.8 までの操作を行う。

Firewall がインストールされていない、もしくは起動していない場合、以下の表示になる。

（起動していない場合）not running

（インストールされていない場合）bash: firewall-cmd: command not found

この場合、以下の 4.1.1.2～4.1.1.8 までの操作は不要である。

4.1.1.2. REST 要求（FC、起動通知、Trap 通知）の接続許可

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行し、REST 要求で使用するポートの接続許可を行う。
(本ドキュメントに記載するポートはデフォルト値の為、コンフィグと共に変更も可能)

【firewall-cmd --permanent --add-port=18080/tcp】[Enter]

4.1.1.3. PostgreSQL の接続許可

＜実行ホスト：DB＞

以下のコマンドを実行して、PostgreSQL に使用するポートの接続許可を行う。

【firewall-cmd --permanent --add-port=5432/tcp】[Enter]

4.1.1.4. SNMPTrap の接続許可

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行して、SNMPTrap に使用するポートの接続許可を行う。

【firewall-cmd --permanent --add-port=162/udp】[Enter]

4.1.1.5. syslog 転送受付許可

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行して、syslog 転送受付に用いるポートの接続許可を行う。

【firewall-cmd --permanent --add-port=514/tcp】[Enter]

【firewall-cmd --permanent --add-port=514/udp】[Enter]

4.1.1.6. Pacemaker/Corosync の接続許可

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行して、Pacemaker/Corosync に使用するポートの接続許可を行う。

【firewall-cmd --permanent --add-service=high-availability】[Enter]

【cp -p /usr/lib/firewalld/services/high-availability.xml /etc/firewalld/services/】[Enter]

4.1.1.7. tftpd の接続許可

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行して、tftpd に使用するポートの接続許可を行う。

【firewall-cmd --permanent --add-service=tftp】[Enter]

【cp -p /usr/lib/firewalld/services/tftp.xml /etc/firewalld/services/】[Enter]

4.1.1.8. httpd の接続許可

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行して、httpd に使用するポートの接続許可を行う。

【firewall-cmd --permanent --add-service=http】[Enter]

【cp -p /usr/lib/firewalld/services/http.xml /etc/firewalld/services/】[Enter]

上記、4.1.1 ファイアウォール設定が完了しだい、以下のコマンドを実行し、
設定を反映する。

＜実行ホスト：ACT/SBY/DB＞

【firewall-cmd --reload】[Enter]

【systemctl restart firewalld】[Enter]

現在の設定情報を以下のコマンドを実行し、確認する。（網掛け部分を特に確認する）

【firewall-cmd --list-all】[Enter]

＜実行ホスト：ACT/SBY＞

```
public (default, active)
  interfaces:
  sources:
  services: dhcpv6-client high-availability http ssh tftp
  ports: 514/tcp 162/udp 514/udp 18080/tcp
  masquerade: no
```

forward-ports:
icmp-blocks:
rich rules:

＜実行ホスト：DB＞

public (default, active)
interfaces:
sources:
services: dhcpv6-client ssh
ports: 5432/tcp
masquerade: no
forward-ports:
icmp-blocks:
rich rules:

4.1.1.9. SELinux の設定

＜実行ホスト：ACT/SBY＞

以下のコマンドで現在の設定を確認する。

【getenforce】[Enter]

上記コマンドの結果が「Enfocing」の場合、以下のコマンドを実行して SELinux 設定を変更する。

【setenforce 0】[Enter]

【vi /etc/selinux/config】[Enter]（網掛け部分に修正）

～（略）～
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
# enforcing - SELinux security policy is enforced.
# permissive - SELinux prints warnings instead of enforcing.
# disabled - No SELinux policy is loaded.
SELINUX=disabled
～（略）～

4.1.2. Java のインストール、及び設定

4.1.2.1. インストール

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行し、jdk をインストールする。

【cd ~/setup/ec_main/installer/java.v8u92】[Enter]

【rpm -ivh jdk-8u92-linux-x64.rpm】[Enter]

4.1.2.2. Java バージョンの確認

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行して、バージョンを確認する。

【java -version】[Enter]

インストールが正常に行えていた場合は、以下が表示される。

```
java version "1.8.0_92"
Java(TM) SE Runtime Environment (build 1.8.0_92-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.92-b14, mixed mode)
```

4.1.2.3. profile の修正

＜実行ホスト：ACT/SBY＞

以下のファイルに JRE_HOME の環境変数を追加する。（網掛け部分を追記）

【vi /etc/profile】[Enter]

```
～（略）～
JRE_HOME=/usr/java/jdk1.8.0_92/
export JRE_HOME
```

4.1.3. Net-SNMP のインストール、及び設定

4.1.3.1. インストール

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行し、Net-SNMP をインストールする。

【cd ~/setup/ec_main/installer/snmptrapd.v5.7.2】[Enter]
【rpm -ivh glibc-common-2.17-105.el7.x86_64.rpm】[Enter]
【rpm -ivh glibc-2.17-105.el7.x86_64.rpm】[Enter]
【rpm -ivh perl-*】[Enter]
【rpm -ivh openssl098e-0.9.8e-29.el7.centos.3.x86_64.rpm】[Enter]
【rpm -ivh net-snmp-libs-5.7.2-24.el7_2.1.x86_64.rpm】[Enter]
【rpm -ivh lm_sensors-libs-3.3.4-11.el7.x86_64.rpm】[Enter]
【rpm -ivh net-snmp-agent-libs-5.7.2-24.el7_2.1.x86_64.rpm】[Enter]
【rpm -ivh net-snmp-5.7.2-24.el7_2.1.x86_64.rpm】[Enter]

※既にインストール済みの出力結果も問題なしとする。

4.1.3.2. SNMPTrap 設定ファイルの修正

＜実行ホスト：ACT/SBY＞

SNMPTrap 設定ファイル(/etc/snmp/snmptrapd.conf)に以下の内容を追加する。(網掛け部分を追記)
下記の$EC_HOME/は 4.2.3 スクリプト配置と同じパスを指定すること。

【vi /etc/snmp/snmptrapd.conf】[Enter]

```
～（略）～
authCommunity log,execute public
#Linkdown の trap 受信
traphandle .1.3.6.1.6.3.1.1.5.3 $EC_HOME/ec_main/bin/linkdown.sh
#Linkup の trap 受信
traphandle .1.3.6.1.6.3.1.1.5.4 $EC_HOME/ec_main/bin/linkup.sh
```

SNMPTrap サービスファイル(/usr/lib/systemd/system/snmptrapd.service)に以下の内容を追加する。

【vi /usr/lib/systemd/system/snmptrapd.service】[Enter] (網掛け部分を追記)

```
～（略）～
[Service]
Type=notify
Environment=OPTIONS="-Lsd"
EnvironmentFile=-/etc/sysconfig/snmptrapd
ExecStart=/usr/sbin/snmptrapd -On $OPTIONS -f
ExecReload=/bin/kill -HUP $MAINPID
～（略）～
```

4.1.4. rsyslog の設定

4.1.4.1. rsyslog 設定ファイルの修正

＜実行ホスト：ACT/SBY＞

rsyslog 設定ファイル(/etc/rsyslog.conf)を以下の内容に修正する。（網掛け部分に修正）

下記の「装置 CIDR」は装置のセグメントの CIDR 表記を記載する。

【vi /etc/rsyslog.conf】[Enter]

修正前

～（略）～

```
# Provides UDP syslog reception
#$ModLoad imudp
#$UDPServerRun 514

# Provides TCP syslog reception
#$ModLoad imtcp
#$InputTCPServerRun 514
```

修正後

～（略）～

```
# Provides UDP syslog reception
$ModLoad imudp
$UDPServerRun 514
$AllowedSender UDP, 127.0.0.1, 装置 CIDR

# Provides TCP syslog reception
$ModLoad imtcp
$InputTCPServerRun 514
$AllowedSender TCP, 127.0.0.1, 装置 CIDR
```

～（略）～

以下ファイル末尾に追記

```
$template hostip, "%fromhost-ip%"
```

4.1.5. NTP のインストール、及び設定

4.1.5.1. インストール

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行し、ntp をインストールする。

【cd ~/setup/ec_main/installer/ntp.v4.2/】[Enter]

【rpm -ivh autogen-libopts-5.18-5.el7.x86_64.rpm】[Enter]

【rpm -ivh ntpdate-4.2.6p5-22.el7.centos.x86_64.rpm】[Enter]

【rpm -ivh ntp-4.2.6p5-22.el7.centos.x86_64.rpm】[Enter]

4.1.5.2. drift ファイルの作成

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行し、空の drift ファイルを作成する。

【touch /var/lib/ntp/drift】[Enter]

4.1.5.3. NTP 設定ファイルの修正

＜実行ホスト：ACT/SBY＞

NTP 設定ファイル(/etc/ntp.conf)に以下の内容(網掛け部分)を追加する。

【vi /etc/ntp.conf】[Enter]

～（略）～

restrict default nomodify notrap nopeer noquery

restrict default ignore

～（略）～

#restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap

restrict xxx.xxx.xxx.xxx noquery nomodify
server xxx.xxx.xxx.xxx iburst

～（略）～

4.1.5.4. NTP サーバとの同期

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行し、NTP が起動していないことを確認する。

【systemctl status ntpd.service】[Enter]

<NTP 未起動時の出力>

～（略）～
Active: inactive (dead)
～（略）～

<NTP 起動時の出力>

～（略）～
Active: active (running)

～（略）～

NTP が起動している場合、以下のコマンドを実行して NTP を停止する。

【systemctl stop ntpd.service】[Enter]

NTP サーバ(IP アドレス：xxx.xxx.xxx.xxx)と時刻を合わせる。

【ntpdate xxx.xxx.xxx.xxx】[Enter]

4.1.5.5. NTP の再起動

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行し、NTP を再起動する。

【systemctl restart ntpd.service】[Enter]

【systemctl enable ntpd.service】[Enter]

以下のコマンドを実行し、NTP サーバとの同期を確認する。

【ntpq -p】[Enter]

<同期成功時出力例>

```
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*xxx.xxx.xxx.xxx LOCAL(0)        11 u   55   64  377    0.130   -0.017   0.017
```

4.1.6. PostgreSQL のインストール、及び設定

4.1.6.1. インストール

＜実行ホスト：DB＞

以下のコマンドを実行し、postgresql をインストールする。

【cd ~/setup/ec_main/installer/postgresql.v9.3.13】[Enter]

【rpm -ivh libxslt-1.1.28-5.el7.x86_64.rpm】[Enter]

【rpm -ivh uuid-1.6.2-26.el7.x86_64.rpm】[Enter]

【rpm -ivh postgresql93-libs-9.3.13-1PGDG.rhel7.x86_64.rpm】[Enter]

【rpm -ivh postgresql93-9.3.13-1PGDG.rhel7.x86_64.rpm】[Enter]

【rpm -ivh postgresql93-server-9.3.13-1PGDG.rhel7.x86_64.rpm】[Enter]

【rpm -ivh postgresql93-devel-9.3.13-1PGDG.rhel7.x86_64.rpm】[Enter]

【rpm -ivh postgresql93-contrib-9.3.13-1PGDG.rhel7.x86_64.rpm】[Enter]

4.1.6.2. PostgreSQL のコンフィグ修正

＜実行ホスト：DB＞

以下の内容に修正する。

【vi /var/lib/pgsql/.bash_profile】[Enter] （網掛け部分を修正および追記する）

PGDATA=/usr/local/pgsql/9.3/data

export PGDATA

export PATH=$PATH:/usr/pgsql-9.3/bin

【source /var/lib/pgsql/.bash_profile】[Enter]

4.1.6.3. データベース作成、及び権限付与

＜実行ホスト：DB＞

以下のコマンドを実行して、データベースのインストール先フォルダを作成する。

【cd /usr/local/】[Enter]

【mkdir -pm 777 /usr/local/pgsql/9.3】[Enter]

【chown -R postgres:postgres pgsql】[Enter]

postgres ユーザで以下のコマンドを実行してデータベースを作成する。

【su - postgres】[Enter]

【cd /usr/local/pgsql/9.3/】[Enter]

【mkdir -m 700 data】[Enter]

【initdb --encoding=UTF8 --no-locale --pgdata=/usr/local/pgsql/9.3/data --auth=ident】[Enter]

【pg_ctl -D /usr/local/pgsql/9.3/data -l logfile start】[Enter]

【psql -c "alter user postgres with password ""】[Enter]

【psql】[Enter]

【create role root login createdb password ''; 】[Enter]

【¥q】[Enter]

【pg_ctl -D /usr/local/pgsql/9.3/data -l logfile stop】[Enter]

【exit】[Enter]

ルートユーザで以下のコマンドを実行する。

【systemctl enable postgresql-9.3】[Enter]

【systemctl daemon-reload】[Enter]

4.1.6.4. データベースのコンフィグ修正

＜実行ホスト：DB＞

以下の内容に修正する。

【vi /usr/local/pgsql/9.3/data/postgresql.conf】[Enter]

修正前

～（略）～

#listen_addresses = 'localhost'
#port = 5432

～（略）～

修正後

～（略）～

listen_addresses = '*'
port = 5432

～（略）～

以下の内容に修正する。（網掛け部分は許可するサーバのセグメントを記載）

【vi /usr/local/pgsql/9.3/data/pg_hba.conf】[Enter]

修正前

～（略）～

```
# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     peer
# IPv4 local connections:
host    all             all             127.0.0.1/32            ident
# IPv6 local connections:
host    all             all             ::1/128                 ident
# Allow replication connections from localhost, by a user with the
# replication privilege.
#local   replication     postgres                                peer
#host    replication     postgres        127.0.0.1/32            ident
#host    replication     postgres        ::1/128                 ident
```

修正後

～（略）～

```
# TYPE  DATABASE        USER            ADDRESS                 METHOD
```

```
# "local" is for Unix domain socket connections only
#local     all                 all                                         peer
# IPv4 local connections:
#host      all                 all                 127.0.0.1/32            ident
# IPv6 local connections:
#host      all                 all                 ::1/128                 ident
# Allow replication connections from localhost, by a user with the
# replication privilege.
#local     replication         postgres                                    peer
#host      replication         postgres            127.0.0.1/32            ident
#host      replication         postgres            ::1/128                 ident

local      all                 postgres                                    peer
local      all                 all                                         trust
host       all                 all                 EC の CIDR              trust
host       all                 all                 127.0.0.1/32            trust
```

以下の内容に修正する。（網掛け部分を修正する）

【vi /usr/lib/systemd/system/postgresql-9.3.service】[Enter]

```
～（略）～
# Location of database directory
Environment=PGDATA=/usr/local/pgsql/9.3/data/
～（略）～
```

4.1.6.5. データベースの再起動

＜実行ホスト：DB＞

以下のコマンドを postgres ユーザで実行する。

【systemctl daemon-reload】[Enter]

【systemctl start postgresql-9.3】[Enter]

4.1.7. DHCP のインストール、及び設定

4.1.7.1. インストール

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行し、DHCP をインストールする。

【cd ~/setup/ec_main/installer/dhcp.v4.2.5】[Enter]

【rpm -ihv dhcp-4.2.5-42.el7.centos.x86_64.rpm】[Enter]

4.1.7.2. systemctl のユニットファイル作成

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行し、systemctl のユニットファイルを作成する。

【cp -p /usr/lib/systemd/system/dhcpd.service /etc/systemd/system/】[Enter]

【vi /etc/systemd/system/dhcpd.service】[Enter] （網掛け部分を追記する）

～（略）～

Type=notify

ExecStart=/usr/sbin/dhcpd -f -cf /etc/dhcp/dhcpd.conf -user dhcpd -group dhcpd --no-pid 「インタフェース名」

～（略）～

DHCP 設定ファイルに以下の内容を追加する。

【vi /etc/sysconfig/dhcpd】[Enter] （網掛け部分を追記する）

～（略）～

DHCPDARGS=「インタフェース名」

【vi /etc/dhcp/dhcpd.conf.template.NCS5001】[Enter] （網掛け部分を追記する）

```
# DHCP server general settings
subnet $$MANAGEMENTNETWORKADDRESS$$ netmask $$MANAGEMENTSUBNETMASK$$ {
  range $$MANAGEMENTRANGESTART$$ $$MANAGEMENTRANGEEND$$;
  option subnet-mask $$MANAGEMENTSUBNETMASK$$;
  default-lease-time 600;
  deny unknown-clients;
  max-lease-time 7200;
}
###########
# host
###########
# NCS5001
group {
  filename "$$INITIALCONFIG$$";
  host ncs5k {
    hardware ethernet $$MACADDRESS$$;
    fixed-address $$MANAGEMENTADDRESS$$;
  }
}
```

【vi /etc/dhcp/dhcpd.conf.template.QFX5100】[Enter] （網掛け部分を追記する）

```
# For QFX zero touch provisioning
```

```
option space QFX;
option QFX.config-file-name code 1 = text;
option QFX.image-file-type code 2 = text;
option QFX.transfer-mode code 3 = text;
option QFX.alt-image-file-name code 4= text;
option QFX-encapsulation code 43 = encapsulate QFX;

# DHCP server general settings
subnet $$MANAGEMENTNETWORKADDRESS$$ netmask $$MANAGEMENTSUBNETMASK$$ {
  range $$MANAGEMENTRANGESTART$$ $$MANAGEMENTRANGEEND$$;
  option subnet-mask $$MANAGEMENTSUBNETMASK$$;
  default-lease-time 600;
  deny unknown-clients;
  max-lease-time 7200;
}
##########
# host
##########
# QFX5100
host QFX5100 {
  hardware ethernet $$MACADDRESS$$;
  fixed-address $$MANAGEMENTADDRESS$$;
  option tftp-server-name "$$TFTPHOSTNAME$$";
  option host-name "$$HOSTNAME$$";
  option log-servers $$LOGSERVERADDRESS$$;
  option QFX.transfer-mode "tftp";
  option QFX.config-file-name "$$INITIALCONFIG$$";
}
```

【vi /etc/dhcp/dhcpd.conf.template.QFX5200】[Enter] （網掛け部分を追記する）

```
# For QFX zero touch provisioning
option space QFX;
option QFX.config-file-name code 1 = text;
option QFX.image-file-type code 2 = text;
option QFX.transfer-mode code 3 = text;
option QFX.alt-image-file-name code 4= text;
option QFX-encapsulation code 43 = encapsulate QFX;

# DHCP server general settings
subnet $$MANAGEMENTNETWORKADDRESS$$ netmask $$MANAGEMENTSUBNETMASK$$ {
  range $$MANAGEMENTRANGESTART$$ $$MANAGEMENTRANGEEND$$;
  option subnet-mask $$MANAGEMENTSUBNETMASK$$;
  default-lease-time 600;
  deny unknown-clients;
  max-lease-time 7200;
}
##########
# host
##########
# QFX5200
host QFX5200-3 {
  hardware ethernet $$MACADDRESS$$;
  fixed-address $$MANAGEMENTADDRESS$$;
  option tftp-server-name "$$TFTPHOSTNAME$$";
  option host-name "$$HOSTNAME$$";
```

```
    option log-servers $$LOGSERVERADDRESS$$;
    option QFX.transfer-mode "http";
    option QFX.config-file-name "$$INITIALCONFIG$$";
}
```

4.1.8. tftpd のインストール、及び設定

4.1.8.1. tftpd のインストール

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行し、tftpd をインストールする。

【cd ~/setup/ec_main/installer/tftp.v5.2】[Enter]

【rpm -ihv xinetd-2.3.15-12.el7.x86_64.rpm】[Enter]

【rpm -ihv tftp-5.2-12.el7.x86_64.rpm】[Enter]

【rpm -ihv tftp-server-5.2-12.el7.x86_64.rpm】[Enter]

4.1.8.2. tftpd の設定

＜実行ホスト：ACT/SBY＞

tftp 設定ファイル(/etc/xinet.d/tftp)を以下の内容(網掛け部分)に修正する。

【vi /etc/xinetd.d/tftp】[Enter]

修正前

～（略）～
 disabled = yes
～（略）～

修正後

～（略）～
 disabled = no
～（略）～

xinted を起動する。

【systemctl start xinetd】[Enter]

4.1.9.　httpd のインストール

4.1.9.1.　httpd のインストール

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行し、httpd をインストールする。

【cd ~/setup/ec_main/installer/httpd.v2.4.6】[Enter]

【rpm -ihv apr-1.4.8-3.el7.x86_64.rpm】[Enter]

【rpm -ihv apr-util-1.5.2-6.el7.x86_64.rpm】[Enter]

【rpm -ihv mailcap-2.1.41-2.el7.noarch.rpm】[Enter]

【rpm -ihv httpd-tools-2.4.6-40.el7.centos.4.x86_64.rpm】[Enter]

【rpm -ihv httpd-2.4.6-40.el7.centos.4.x86_64.rpm】[Enter]

【rpm -ihv httpd-manual-2.4.6-40.el7.centos.4.noarch.rpm】[Enter]

【rpm -ihv mod_ssl-2.4.6-40.el7.centos.4.x86_64.rpm】[Enter]

httpd を起動する。

【systemctl start httpd】[Enter]

【systemctl enable httpd】[Enter]

4.1.10. Pacemaker のインストール、及び設定

4.1.10.1. Pacemaker のインストール

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行し、インストーラを起動し、インストールを行う。

【cd ~/setup/ec_main/installer/pacemaker.v1.1.14-1.1/】[Enter]

【sh pacemaker_install.sh】[Enter]

補足：インストール中「警告: pssh-2.3.1-5.el7.noarch.rpm: ヘッダー V3 RSA/SHA256 Signature、鍵 ID 352c64e5: NOKEY」といった形で鍵がないことを理由に画面へ警告が表示されるが、インストールは進行するので問題ない。

4.1.10.2. Pacemaker のインストール確認

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行して、Corosync のバージョンを確認する。

【corosync -version】[Enter]

インストールが正常に完了した場合は、以下が表示される。

Corosync Cluster Engine, version '2.3.5'

Copyright (c) 2006-2009 Red Hat, Inc.

以下のコマンドを実行して、pcs のバージョンを確認する。

【pcs --version】[Enter]

インストールが正常に完了した場合は、以下が表示される。

0.9.143

以下のコマンドを実行して、Pacemaker のバージョンを確認する。

【crmadmin --version】[Enter]

インストールが正常に完了した場合は、以下が表示される。

Pacemaker 1.1.14-1.el7

Written by Andrew Beekhof

以下のコマンドを実行して、crm のバージョンを確認する。

【crm --version】[Enter]

インストールが正常に完了した場合は、以下が表示される。

2.1.5-1.el7 (Build unknown)

以下のコマンドを実行して、用いるリソースエージェントがインストールされているかを確認する。

【ls /lib/ocf/resource.d/pacemaker/】[Enter]

インストールが正常に完了した場合は、以下が表示される。

diskd

以下のコマンドを実行して、用いるリソースエージェントがインストールされているかを確認する。

【ls /lib/ocf/resource.d/heartbeat/】[Enter]

インストールが正常に完了した場合は、以下が表示される。

VIPcheck、IPaddr2

4.1.10.3. host の設定

＜実行ホスト：ACT/SBY＞

稼働系と待機系で用いるノードのホストを登録する。

この作業は稼働系、待機系両ノードで実施する。

以下のコマンドを実行して、hosts ファイルを開く

【vi /etc/hosts】[Enter]

文末に以下を追加する

(待機系のインターコネクト用 IP アドレス) (待機系のホスト名)

(稼働系のインターコネクト用 IP アドレス) (稼働系のホスト名)

追加後、保存する

[esc]【:wq】[Enter]

Hosts の設定を確認する。

【ping 「待機系ホスト名」】[Enter]

PING 「待機系ホスト名」(待機系のインターコネクト用 IP アドレス) 56(84) bytes of data.

64 bytes from 「待機系ホスト名」(待機系のインターコネクト用 IP アドレス): icmp_seq=1 ttl=64 time=0.166 ms

【ping 「稼働系ホスト名」】[Enter]

PING 「稼働系ホスト名」(稼働系のインターコネクト用 IP アドレス) 56(84) bytes of data.

64 bytes from 「稼働系ホスト名」(稼働系のインターコネクト用 IP アドレス): icmp_seq=1 ttl=64 time=0.166 ms

上記のように IP アドレス、ホスト名が表示されない場合、/etc/hosts の設定を再確認する。

4.1.10.4. パスワードの設定

＜実行ホスト：ACT/SBY＞

再起動後、corosync によるサーバ間通信の認証で使用される"hacluster"ユーザのパスワードを設定する。パッケージインストール時に自動的にユーザが作成されているため、パスワード変更のみ行う。パスワードはクラスタを構成するすべてのサーバにて同一とする。

この作業は稼働系、待機系両ノードで実施する。

以下のコマンドを実行して、パスワード変更を行う

【passwd hacluster】[Enter]

コマンド実行後、以下の文章が表示されるので、新しいパスワードの入力を行う

Changing password for user hacluster.

New password:

【(新しいパスワード)】[Enter]

[Enter]押下後、確認のため再度パスワードを入力するよう、指示される。

Retype new password:

同じパスワードを入力する

【(新しいパスワード)】[Enter]

パスワードが同一であれば、以下の文章が表示され、パスワードの変更が完了する。

passwd: all authentication tokens updated successfully.

4.1.10.5. pscd サービスの設定

＜実行ホスト：ACT/SBY＞

pcsd サービスを起動および有効化する。pcsd は pacemaker や corosync とは独立したサービスであり、これが起動していないとクラスタ構成時に使用する pcs コマンドが使えない。

この作業は稼働系、待機系の両ノードで実施する。

以下のコマンドを実行して、pscd サービスの有効化を行う。

【systemctl start pcsd】[Enter]

【systemctl enable pcsd】[Enter]

enable 操作後、以下のメッセージが画面に表示される

Created symlink from /etc/systemd/system/multi-user.target.wants/pcsd.service to /usr/lib/systemd/system/pcsd.service.

4.1.10.6. pscd サービス状況の確認

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行して、pcsd のサービス状況を確認する。

この作業は稼働系、待機系の両ノードで実施する。

【systemctl status pcsd】[Enter]

正しく起動できていれば、以下のメッセージが画面に表示される

```
  Loaded: loaded (/usr/lib/systemd/system/pcsd.service; disabled; vendor preset: disabled)
  Active: active (running) since *** 2016-**-** **:**:** UTC; **s ago
Main PID: **** (pcsd)
  CGroup: /system.slice/pcsd.service
          ├**** /bin/sh /usr/lib/pcsd/pcsd start
          ├**** /bin/bash -c ulimit -S -c 0 >/dev/null 2>&1 ; /usr/bin/ruby -I/usr/lib/...
          └**** /usr/bin/ruby -I/usr/lib/pcsd /usr/lib/pcsd/ssl.rb
```

4.1.10.7. ノード認証

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行して、"pcs cluster auth"コマンドにより互いのノードの認証を行う。

これにより、corosync によるサーバ間通信が可能となる。

この作業は稼働系、待機系の両ノードで実施する。

【pcs cluster auth (稼働系ノード名) (待機系のノード名)　-u hacluster -p (hacluster のパスワード) --force】[Enter]

正しく認証できれば、以下のメッセージが画面に表示される

(待機系ホスト名 or 稼働系ホスト名): Authorized

(待機系ホスト名 or 稼働系ホスト名): Authorized

4.1.10.8. 初期クラスタ作成

＜実行ホスト：ACT＞

以下のコマンドを実行して、"pcs cluster setup"コマンドにより、初期クラスタの作成を行う。

この作業は稼働系のノードのみで行う。

【pcs cluster setup --name (設定するクラスタ名) (稼働系のノード名) (待機系のノード名)】

正しくクラスタが生成できれば、以下のメッセージが画面に表示される

Shutting down pacemaker/corosync services...

Redirecting to /bin/systemctl stop　pacemaker.service

Redirecting to /bin/systemctl stop　corosync.service

Killing any remaining services...

Removing all cluster configuration files...

(稼働系のノード名): Succeeded

(待機系のノード名): Succeeded

Synchronizing pcsd certificates on nodes (稼働系のノード名), (待機系のノード名)...

(稼働系のノード名): Succeeded

(待機系のノード名): Succeeded

Restaring pcsd on the nodes in order to reload the certificates...

(稼働系のノード名): Succeeded

(待機系のノード名): Succeeded

4.1.10.9. ノード同期

＜実行ホスト：ACT＞

以下のコマンドを実行して、両ノードの同期を開始し、クラスタとして成り立たせる。

この作業は稼働系のノードのみで行う。

【pcs cluster start --all】[Enter]

正しくクラスタが開始できれば、以下のメッセージが画面に表示される

(稼働系のノード名): Starting Cluster...
(待機系のノード名): Starting Cluster...

4.1.10.10. ノード間通信状況確認

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行して、"corosync-cfgtool -s"コマンドにより、corosync によるノード間通信の状況を確認する。

この作業は稼働系、待機系、両ノードで行う。

【corosync-cfgtool -s】[Enter]

正しくクラスタが開始している場合、以下のメッセージが画面に表示される。

"status"が"active"かつ"no faults"であれば、問題なく通信が行えている。

```
Printing ring status.
Local node ID (1 or 0)
RING ID 0
        id      = (稼働系 or 待機系の IP アドレス)
        status  = ring 0 active with no faults
```

4.1.10.11. クラスタ状態確認

＜実行ホスト：ACT or SBY＞

以下のコマンドを実行して、"pcs status"コマンドでクラスタ状態を確認する。

この作業は待機系、稼働系のどちらかで行えば良い。

【pcs status】[Enter]

コマンド実行後、画面にクラスタの状態が表示されるが、表示最上部が以下のように、STONITH についての警告が表示される。

```
Cluster name: (設定したクラスタ名)
WARNING: no stonith devices and stonith-enabled is not false
Last updated: Thu Oct 13 02:06:57 2016          Last change: Thu Oct 13 02:06:49 2016
```

今回の環境では STONITH は利用できないため、STONITH の利用を FALSE に設定する。

【pcs property set stonith-enabled=false】[Enter]

最後にもう一度クラスタの状態を確認する。

【pcs status】[Enter]

コマンド実行後、画面に、クラスタの状態が表示される。正しく設定できていれば以下のように表示される。

```
Cluster name: eccluster
```

Last updated: WDW MMM DD HH:MM:DD YYYY Last change: WDW MMM DD HH:MM:DD YYYY by root via cibadmin on (稼働系のノード名)
Stack: corosync
Current DC: (待機系のノード名) (version 1.1.14-1.el7-70404b0) - partition with quorum
2 nodes and 0 resources configured

Online: [(稼働系のノード名)　(待機系のノード名)]

Full list of resources:

PCSD Status:
(稼働系のノード名): Online
(待機系のノード名): Online

Daemon Status:
corosync: active/disabled
pacemaker: active/disabled
pcsd: active/enabled

4.1.10.12. pacemaker/corosync サービス有効化

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行して、再起動後も corosync と pacemaker サービスの自動起動を有効にするため、両サービスを enable にする。

この作業は待機系、稼動系の両ノードで行う。

【systemctl enable pacemaker】[Enter]

【systemctl enable corosync】[Enter]

4.2. ECメインモジュールインストール

以下、記載の$EC_HOMEはユーザ指定の任意の場所とする。

下記は/usrをインストールディレクトリとする例。

【export EC_HOME=/usr】[Enter]

4.2.1. ライブラリ配置

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行して付属品内のライブラリファイルを配置する。

【mkdir -p $EC_HOME/ec_main/lib】[Enter]

【cp ~/setup/ec_main/lib/* $EC_HOME/ec_main/lib/】[Enter]

4.2.2. コンフィグ配置

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行して付属品内のコンフィグファイルを配置する。

【mkdir -p $EC_HOME/ec_main/conf】[Enter]

【cp ~/setup/ec_main/conf/* $EC_HOME/ec_main/conf/】[Enter]

以下のコマンドにて、ECメインモジュール設定ファイルを修正する。

【vi $EC_HOME/ec_main/conf/ec_main.conf】[Enter]

※インストールしたサーバに合わせて、コンフィグ値を修正して下さい。

修正内容は「別紙_コンフィグ仕様書.xlsx」を参照。

以下のコマンドにて、Hibernete設定ファイルを修正する。

【vi $EC_HOME/ec_main/conf/hibernate.cfg.xml】[Enter]（網掛け部分を修正）

～（略）～

<property name="connection.url">jdbc:postgresql://「接続先URL(IPアドレスxxx.xxx.xxx.xxx):(ポート番号XXXX)」/msf_ec</property>

～（略）～

以下のコマンドにて、log4j設定ファイルを修正する。

【mkdir -p 「ログ出力先」】[Enter]

【vi $EC_HOME/ec_main/conf/log4j2.xml】[Enter]（網掛け部分を修正）

～（略）～

<Properties>
    <Property name="log-path">「ログ出力先」</Property>
</Properties>

～（略）～

4.2.3. スクリプト配置

＜実行ホスト：ACT/SBY＞

以下のコマンドを実行して付属品内のスクリプトファイルを配置する。

【mkdir -p $EC_HOME/ec_main/bin】[Enter]

【cp ~/setup/ec_main/bin/* $EC_HOME/ec_main/bin/】[Enter]

【chmod 755 $EC_HOME/ec_main/bin/*】[Enter]

【cd $EC_HOME/ec_main/bin/】[Enter]

【ln -s boot.sh boot_fail.sh】[Enter]

【ln -s boot.sh boot_success.sh】[Enter]

以下のファイルに PATH の環境変数を追加する。

【vi /etc/profile】[Enter]

～（略）～

export PATH=$PATH:$EC_HOME/ec_main/bin

以下のコマンドにて環境変数を読み込ませる。

【source /etc/profile】[Enter]

以下のコマンドにてスクリプトの環境定義を修正する。

【vi $EC_HOME/ec_main/bin/ec_ctl.sh】[Enter]（網掛け部分を修正）

～（略）～

環境定義

HOST="xxx.xxx.xxx.xxx"

PORT="yyyyy"

～（略）～

【vi $EC_HOME/ec_main/bin/boot.sh】[Enter]（網掛け部分を修正）

～（略）～

環境定義

HOST="xxx.xxx.xxx.xxx"

PORT="yyyyy"

～（略）～

【vi $EC_HOME/ec_main/bin/linkup.sh】[Enter]（網掛け部分を修正）

～（略）～

環境定義

HOST="xxx.xxx.xxx.xxx"

PORT="yyyyy"

～（略）～

【vi $EC_HOME/ec_main/bin/linkdown.sh】[Enter] （網掛け部分を修正）

～（略）～

環境定義

HOST="xxx.xxx.xxx.xxx"

PORT="yyyyy"

～（略）～

xxx.xxx.xxx.xxx はインストール先のサーバ VIP アドレスを設定すること。

yyyyy は$EC_HOME/ec_main/conf/ec_main.conf の rest_server_port と同じ値にすること。

4.2.4. スキーマ作成

＜実行ホスト：DB＞

以下のコマンドにて、スキーマを作成する。

【createdb msf_ec】[Enter]

以下のコマンドにて、スキーマにテーブルを作成する。

【psql msf_ec < ~/setup/ec_main/database/create_table.sql】[Enter]

4.3.　リソースエージェントの登録と設定

4.3.1.　リソースエージェントの配置

＜実行ホスト：ACT/SBY＞

EC のリソースエージェントを既定のリソースエージェントフォルダへとコピーする。

【mkdir -p $EC_HOME/ec_main/RA】[Enter]

【cp ~/setup/ec_main/RA/* $EC_HOME/ec_main/RA/】[Enter]

【cp $EC_HOME/ec_main/RA/ec /lib/ocf/resource.d/heartbeat/】[Enter]

【cp $EC_HOME/ec_main/RA/snmptrapd /lib/ocf/resource.d/heartbeat/】[Enter]

その後コピーした EC リソースエージェントに実行権を付与する。

【cd /lib/ocf/resource.d/heartbeat/】[Enter]

【chmod 755 ec】[Enter]

【chmod 755 snmptrapd】[Enter]

4.3.2.　crm ファイルの作成

＜実行ホスト：ACT＞

付属品のリソースエージェントの設定を記した ra_config.xlsx ファイルを編集し、必要な情報を更新した上で、csv ファイルに変換し、稼働系上のホームフォルダへと配置する。

csv ファイルを配置したフォルダ上で次のコマンドを実行し、csv ファイルをリソースエージェントに登録するための crm ファイルへと変換する。

【pm_crmgen -o $EC_HOME/ec_main/conf/crm_conf.crm (配置した csv ファイル名).csv】[Enter]

正しく変換できた場合なにも表示されないが、csv ファイルに誤りがあった場合、修正箇所が表示される。

4.3.3.　crm ファイルの投入

＜実行ホスト：ACT＞

次のコマンドでリソースエージェントの登録を行う

【crm configure load update $EC_HOME/ec_main/conf/crm_conf.crm】[Enter]

正しく投入できた場合、何も表示されないため次項において確認を行う。（※VIPcheck については規定よりも短い秒数が設定されていると表示がでるが問題ないので、そのまま続けて良い）

設定に重大な誤りがあった場合、誤りのある部分と警告が表示され、続けて投入するかどうかを Y/N で聞かれる。この表記がでる場合、設定ファイルに誤りがあるため、【N】[Enter]として回答する。

4.3.4.　投入結果の確認

＜実行ホスト：ACT or SBY＞

次のコマンドで投入したリソースエージェントの動作状況を確認する。

【pcs status】[Enter]

正しく投入できた場合、以下のように表示される。

Cluster name: eccluster
Last updated: WDW MMM DD HH:MM:SS YYYY Last change: WDW MMM DD HH:MM:SS YYYY by root via cibadmin on (稼動系のノード名 or 待機系のノード名)
Stack: corosync
Current DC: (稼働系のノード名 or 待機系のノード名) (version 1.1.14-1.el7-70404b0) - partition with quorum
2 nodes and 5 resources configured

Online: [(稼動系のノード名) (待機系のノード名)]

Full list of resources:

 Resource Group: grpEC
     vipCheck (ocf::heartbeat:VIPcheck): Started (稼動系のノード名)
     prmEC (ocf::heartbeat:ec): Started (稼動系のノード名)
     prmIP (ocf::heartbeat:IPaddr2): Started (稼動系のノード名)
 Clone Set: clnDiskd [prmDiskd]
     Started: [(稼動系のノード名)(待機系のノード名)]

PCSD Status:
  (稼動系のノード名): Online
  (待機系のノード名): Online

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled