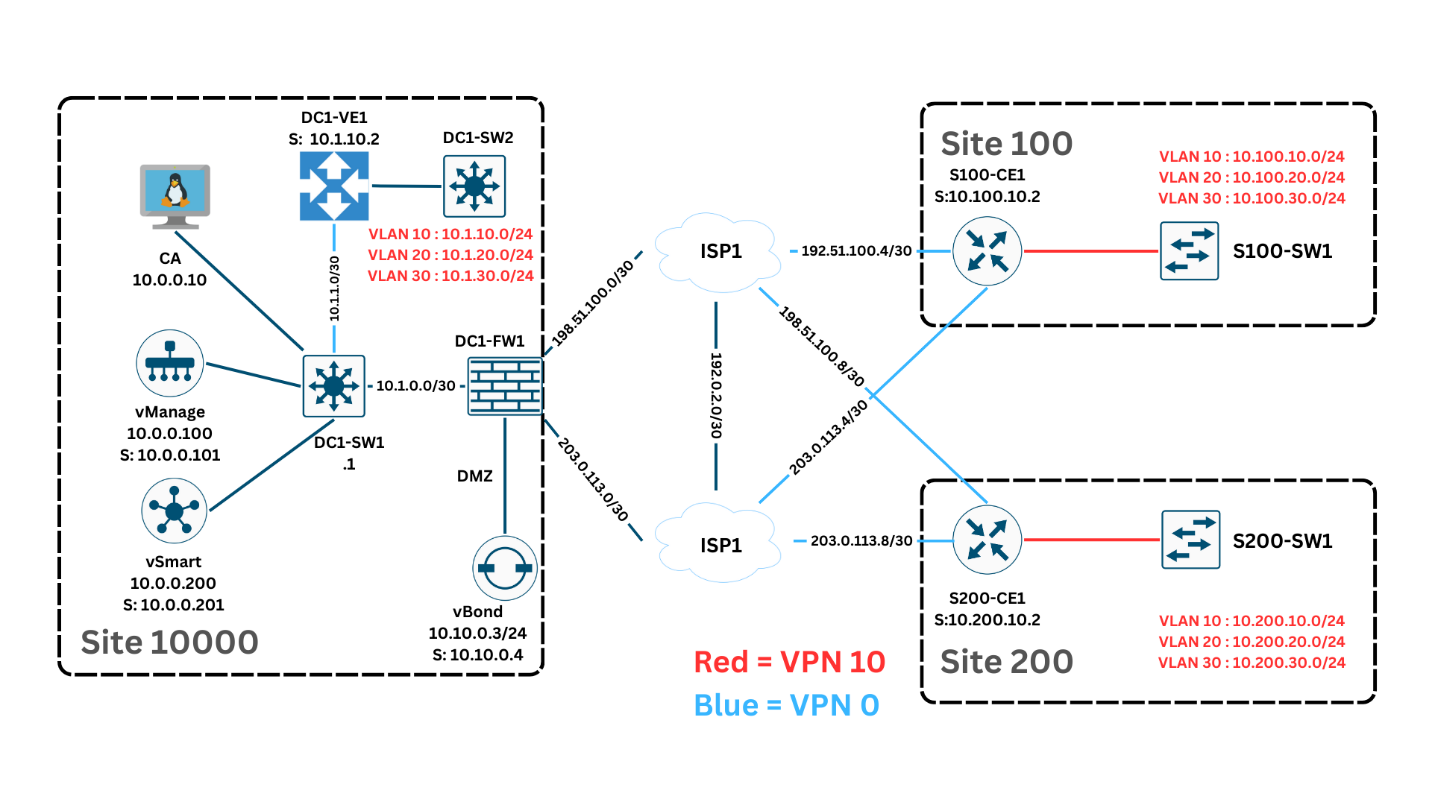
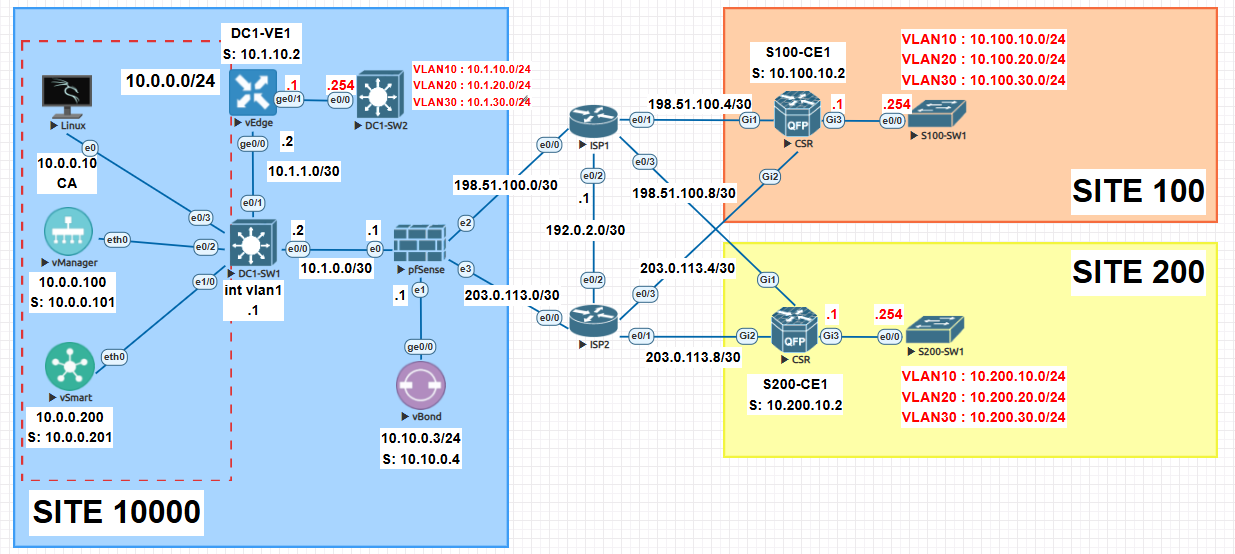
Cisco SD-WAN**:** Basic Configuration Lab



|  |  |
| --- | --- |
| อุปกรณ์ | Image |
| Kali Linux | linux-kali-2023 |
| L3 & L2 Switch | i86bi\_linux\_l2-adventerprisek9-ms.SSA.high\_iron\_20190423 |
| vManage (vtmgmt) | vtmgmt-19.2.0 |
| vSmart (vtsmart) | vtsmart-19.2.0 |
| vBond (vtbond) | vtbond-19.2.0 |
| vEdge (vtedge) | csr1000vng-ucmk9.16.12.1b-sdwan |
| Firewall | pfsense |

Note : Initial Lab Setup on Page 20



Planning sites and system IDs

ใน Cisco SD-WAN เป็นขั้นตอนสำคัญที่ช่วยให้การบริหารจัดการระบบมีความเป็นระเบียบและลดความซับซ้อนในการกำหนดค่าเครือข่าย Site ID เป็นหมายเลข 32 บิตที่ใช้ระบุสถานที่ตั้งของอุปกรณ์ SD-WAN แต่ละตัวภายในองค์กร โดยแต่ละ Site ID จะต้องไม่ซ้ำกันเพื่อให้สามารถระบุอุปกรณ์ใน Control Plane ได้อย่างชัดเจน

System ID เป็นหมายเลข 32 บิตในรูปแบบ dotted decimal ซึ่งต้องอยู่ในช่วงของ valid unicast IP address แม้ว่าค่า System ID จะมีลักษณะคล้ายกับ loopback address หรือ router ID ของโปรโตคอลเราเตอร์ แต่ก็มีข้อกำหนดที่สำคัญคือ ค่านี้จะต้องไม่ถูกกำหนดให้กับอินเทอร์เฟซใด ๆ บนอุปกรณ์ SD-WAN และต้องมีความเป็นเอกลักษณ์ภายใน SD-WAN fabric

สำหรับแนวทางการกำหนดค่า System ID หนึ่งในวิธีที่นิยมใช้กันคือการเลือกหมายเลขที่อยู่ภายในช่วงของ IP subnet ที่อุปกรณ์ SD-WAN ใช้งานอยู่ เช่น หากอุปกรณ์ cEdge S100-CE1 มี IP address ของอินเทอร์เฟซเป็น 10.100.10.1 ก็สามารถกำหนดค่า System ID ให้เป็น 10.100.10.2 ได้ วิธีนี้ช่วยให้สามารถจดจำและจัดการค่าต่าง ๆ ได้ง่ายขึ้น เนื่องจาก System ID และ IP address ของอินเทอร์เฟซมีความสัมพันธ์กันโดยตรง การกำหนดค่าที่เป็นระเบียบและมีรูปแบบที่แน่นอนจะช่วยให้การบริหารจัดการระบบ SD-WAN เป็นไปอย่างมีประสิทธิภาพ ลดความสับสน และช่วยให้สามารถขยายระบบได้ง่ายขึ้นในอนาคต

Planning VPNs

ภายใน Cisco SD-WAN fabric แนวคิดของ **VPNs** มีความคล้ายคลึงกับ **VRFs** ในระบบเราเตอร์แบบดั้งเดิม โดย VPN แต่ละตัวจะถูกแยกจากกันตามค่าเริ่มต้น อุปกรณ์ที่อยู่ภายใน VPN หนึ่งจะไม่สามารถสื่อสารกับอุปกรณ์ที่อยู่ใน VPN อื่นได้ เว้นแต่ว่าจะมีการกำหนด **policy** เพื่ออนุญาตให้สามารถติดต่อกันได้

ลักษณะการทำงานของ VPNs ใน SD-WAN fabric คือ ทุกอินเทอร์เฟซหรือซับอินเทอร์เฟซสามารถเป็นสมาชิกของ VPN ได้เพียงตัวเดียวเท่านั้น หากไม่มีการกำหนดนโยบายเพิ่มเติม อุปกรณ์ทั้งหมดที่อยู่ภายใน VPN เดียวกันจะสามารถสื่อสารกันได้โดยอัตโนมัติ และมีรูปแบบ **full-mesh topology** หรือการเชื่อมต่อแบบ any-to-any ซึ่งเป็นค่าเริ่มต้นของระบบ

อย่างไรก็ตาม สามารถกำหนด **policy** เพื่อเปลี่ยนรูปแบบการเชื่อมต่อของแต่ละ VPN ได้ตามความต้องการ ภายใน SD-WAN fabric เดียวกัน อาจมี VPN หนึ่งที่ใช้ **full-mesh topology** อีก VPN ที่ใช้ **hub-and-spoke** และอีก VPN ที่มีการกำหนด **partial-mesh** เพื่อจำกัดการเชื่อมต่อระหว่างบางไซต์เท่านั้น วิธีนี้ช่วยให้สามารถออกแบบเครือข่ายให้เหมาะสมกับโครงสร้างและความต้องการขององค์กรได้

Planning and implementing templates

เมื่อเริ่มต้นตั้งค่า **SD-WAN environment** ระบบจะมีเพียง **default templates** เท่านั้น โดยยังไม่มีการกำหนด VPNs แบบกำหนดเอง ในสภาวะเริ่มต้นนี้ อุปกรณ์ SD-WAN ทั้งหมดจะมีอินเทอร์เฟซอยู่ใน **VPN 0** (transport VPN) ซึ่งช่วยให้อุปกรณ์สามารถสื่อสารกันในรูปแบบ **full-mesh topology** ได้ แต่หากต้องการให้สามารถส่งข้อมูลของผู้ใช้ผ่าน SD-WAN fabric จำเป็นต้องกำหนด **Service VPN** ซึ่งหมายถึง VPN ใด ๆ ที่ไม่ใช่ VPN 0 หรือ VPN 512

การตั้งค่า VPN และอุปกรณ์ SD-WAN ดำเนินการผ่าน **feature templates** ซึ่งเป็นองค์ประกอบพื้นฐานในการกำหนดค่าต่าง ๆ ของอุปกรณ์ โดย feature templates จะถูกนำมารวมกันเป็น **device templates** และเชื่อมโยงกับอุปกรณ์แต่ละตัว วิธีนี้เป็นแนวทางที่แนะนำในการกำหนดค่าอุปกรณ์ edge แทนการใช้ CLI แบบดั้งเดิม แม้ว่าจะสามารถสร้าง CLI templates ที่รองรับตัวแปรได้ แต่แนวทางนี้ถือว่าเป็นแบบเก่า และผู้ดูแลระบบควรให้ความสำคัญกับการใช้ feature templates เป็นหลัก

เนื่องจาก **templates ส่วนใหญ่จะมีตัวแปร (variables)** ที่ต้องกำหนดค่าเมื่อเชื่อมโยงกับอุปกรณ์ การวางแผนที่รอบคอบจะช่วยให้การนำไปใช้งานมีประสิทธิภาพมากขึ้น หากมีมาตรฐานที่ชัดเจนและลดข้อยกเว้นในการกำหนดค่าจะช่วยลดภาระในการจัดการในอนาคต การมีข้อยกเว้นมากเกินไปจะทำให้เกิด **template sprawl** หรือการกระจายตัวของเทมเพลตมากเกินไป ซึ่งทำให้ระบบซับซ้อนและยุ่งยากต่อการบริหารจัดการ

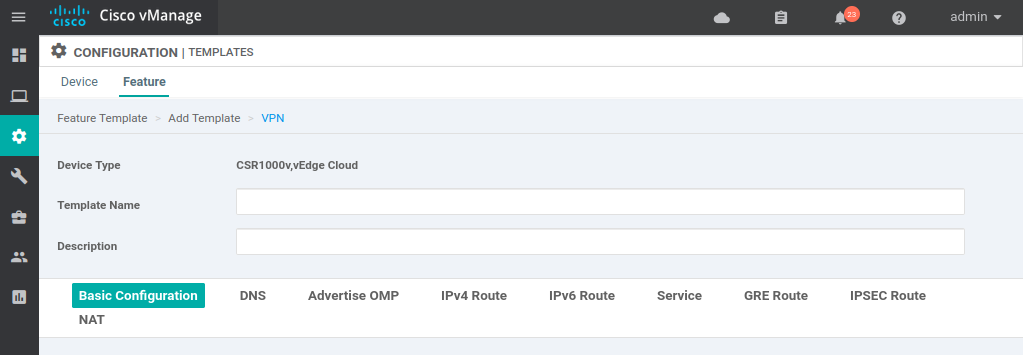
VPN 0 templates

อุปกรณ์ SD-WAN ถูกตั้งค่าเริ่มต้นด้วย **manual ‘skinny’ (หรือ bootstrap) configurations** เพื่อให้สามารถสื่อสารกันได้ในเบื้องต้น ซึ่งหมายความว่า **VPN 0 ถูกกำหนดค่าแบบแมนนวล** ในช่วงแรก อย่างไรก็ตาม เมื่อต้องการเปลี่ยนไปใช้การกำหนดค่าผ่าน **templates** สิ่งแรกที่ต้องทำคือสร้าง **feature template สำหรับ VPN 0** เพื่อให้แน่ใจว่าการเชื่อมต่อระหว่างอุปกรณ์ SD-WAN จะไม่สูญหายเมื่อเริ่มใช้เทมเพลตสำหรับการตั้งค่าอื่น ๆ

ในขั้นตอนนี้ จะมีการเปลี่ยนแปลงค่าต่าง ๆ ให้น้อยที่สุด โดยคงค่าเริ่มต้นไว้มากที่สุดเพื่อสร้างพื้นฐานสำหรับการตั้งค่าขั้นสูงในอนาคต เป้าหมายหลักคือการเริ่มต้นส่งข้อมูลของผู้ใช้ผ่าน SD-WAN fabric โดยใช้การกำหนดค่าที่เรียบง่ายที่สุด

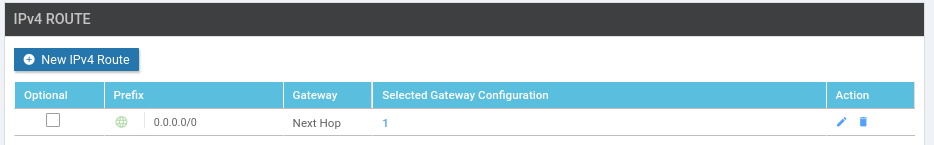
เมื่อสร้างเทมเพลตและเลือกอุปกรณ์หลายตัว รายการของ **sub-templates** ที่สามารถใช้ได้จะถูกจำกัดให้แสดงเฉพาะเทมเพลตที่รองรับบนอุปกรณ์ทุกตัวที่เลือก ตัวอย่างเช่น หากเลือก **CSR1000v และ vEdge Cloud** พร้อมกัน ระบบจะแสดงเฉพาะ **feature templates** ที่สามารถใช้ได้กับทั้งสองอุปกรณ์ รวมถึงเทมเพลตเฉพาะของแต่ละอุปกรณ์ที่สามารถใช้ได้แยกกัน การทำความเข้าใจความแตกต่างนี้จะช่วยให้สามารถออกแบบเทมเพลตให้ครอบคลุมอุปกรณ์ได้อย่างมีประสิทธิภาพมากขึ้น

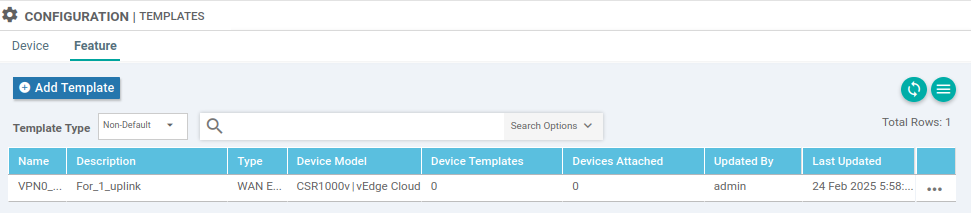
ใน vManage ให้ไปที่ Configuration > Templates > Feature > Add Template จากนั้นเลือก CSR1000v และ vEdge Cloud และคลิกที่ VPN template เพิ่มชื่อและคำอธิบายให้กับ template



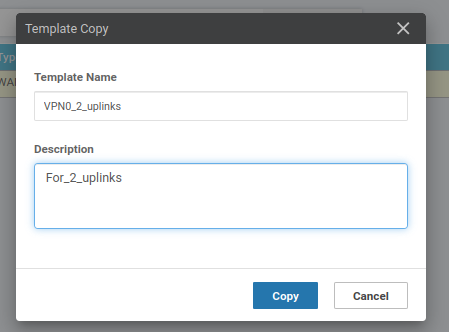
นี่เป็นอีกส่วนหนึ่งที่ต้องใช้การพิจารณาและวางแผนอย่างรอบคอบในสภาพแวดล้อมการผลิต เทมเพลตที่ตั้งชื่อดีจะทำให้การอ้างอิง เข้าใจ และจัดการได้ง่ายขึ้น นี่คือสิ่งที่คุณจะเก่งขึ้นเมื่อมีประสบการณ์ สำหรับการทดลองนี้ เราจะสร้างเทมเพลตสองอันสำหรับ VPN0 โดยเทมเพลตหนึ่งจะถูกเชื่อมต่อกับอุปกรณ์ที่มี uplink เดียว และอีกเทมเพลตหนึ่งสำหรับอุปกรณ์ที่มีสอง uplinks ตั้งชื่อเทมเพลตแรกว่า VPN0\_1\_uplink หลังจากตั้งชื่อเทมเพลตและเพิ่มคำอธิบายแล้ว สิ่งสุดท้ายที่ต้องทำคือกำหนด static default route สำหรับ lab นี้ ให้คลิกที่ปุ่ม New IPv4 Route จากนั้นตั้งค่า Prefix เป็นค่า global 0.0.0.0/0 แล้วคลิก Add Next Hop และกำหนดที่อยู่เป็นตัวแปร Device Specific โดยต้องตั้งชื่อตัวแปรให้ไม่ซ้ำกัน ในที่นี้เลือกใช้ชื่อ default\_next\_hop ค่าของตัวแปรนี้จะถูกกำหนดเมื่อนำเทมเพลตไปผูกกับอุปกรณ์ คลิก Add เพื่อบันทึกค่า next hop จากนั้น

คลิก Add อีกครั้งเพื่อเพิ่ม default route ลงในเทมเพลต และสุดท้ายคลิก Save





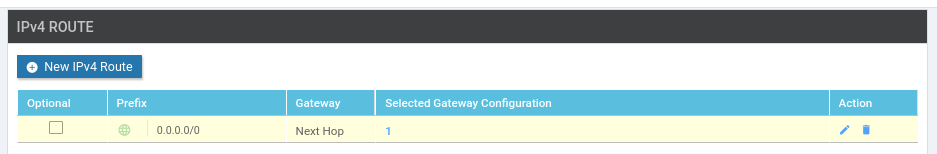
เมื่อใช้เทมเพลต เราจะเห็นถึงประโยชน์ของการจัดการที่เป็นระบบ จากหน้าจอ feature templates ให้คลิกที่ สามจุด ทางขวาของเทมเพลต VPN0\_1\_uplink แล้วเลือก Copy ระบบจะแสดงกล่องโต้ตอบให้แก้ไขชื่อและคำอธิบายของเทมเพลตใหม่ ซึ่งในกรณีนี้สามารถตั้งชื่อเป็น VPN0\_2\_uplinks ได้



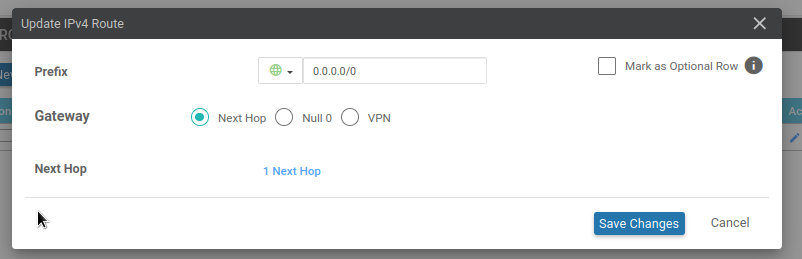
จากนั้นคลิกที่ สามจุด ทางขวาของแถวของเทมเพลตใหม่ แล้วเลือก Edit เพื่อดำเนินการแก้ไขเพิ่มเติม

สำหรับเทมเพลต VPN0\_2\_uplinks สิ่งเดียวที่ต้องเปลี่ยนคือการเพิ่ม next-hop ตัวที่สอง สำหรับ default route ของอุปกรณ์ที่มี dual uplinks ในแต่ละ VPN template สามารถกำหนดค่าเส้นทางเดียวกันได้เพียงครั้งเดียว แต่สามารถมีหลายเส้นทางในเทมเพลตเดียวกัน ใน Cisco IOS CLI แบบดั้งเดิม เราจะสร้าง static default routes สองรายการโดยใช้ next-hop ที่แตกต่างกัน แต่ใน VPN template เรากำหนดเพียง static default route เดียว แล้วระบุ next-hops สองค่า แทน

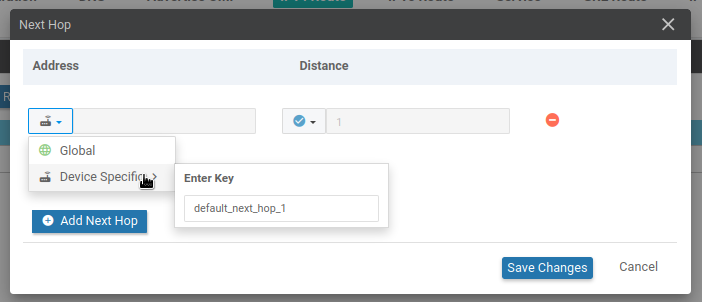
ให้คลิกไอคอน pencil (ดินสอ) ใต้ Action ของ default route



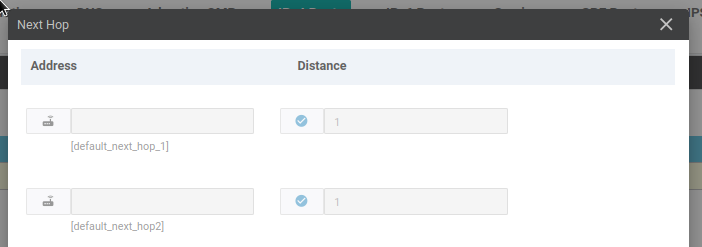
จากนั้นคลิกที่ 1 Next Hop เพื่อแก้ไขค่าของ next-hop



เนื่องจากชื่อตัวแปรต้องไม่ซ้ำกัน ให้เปลี่ยนชื่อ next-hop ปัจจุบัน เป็นชื่อใหม่ที่สามารถจดจำและอ้างอิงได้ง่ายเมื่อนำเทมเพลตไปใช้กับอุปกรณ์ ในที่นี้เลือกใช้ชื่อ default\_next\_hop\_1



จากนั้นคลิก Add Next Hop และกำหนดตัวแปรใหม่สำหรับ next-hop ตัวที่สอง โดยตั้งชื่อเป็น default\_next\_hop\_2



หลังจากกำหนดค่าเสร็จแล้ว คลิก Save Changes สองครั้ง และสุดท้ายคลิก Update เพื่อบันทึกการเปลี่ยนแปลงทั้งหมด

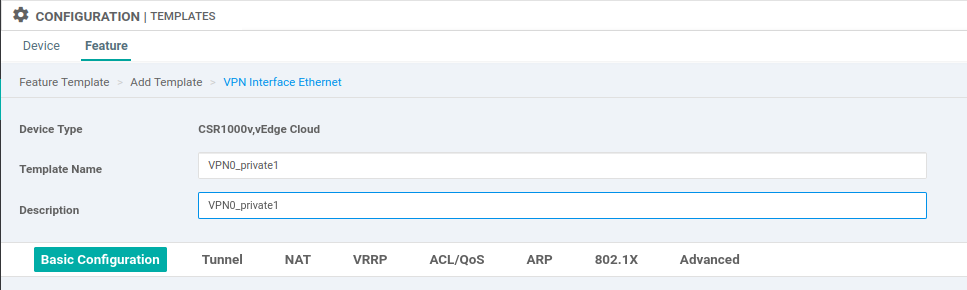
VPN Interface templates

VPN templates จะกำหนด VPN เอง แต่เรายังต้องการ interface เพื่อต่อเชื่อม VPN Interface templates จะอธิบายการกำหนดค่าของ physical หรือ logical interfaces ซึ่งเป็นอีกพื้นที่หนึ่งที่ต้องมีการวางแผนและพิจารณาให้เหมาะสมกับสภาพแวดล้อม เทมเพลตส่วนใหญ่ควรครอบคลุมตัวเลือกที่กว้างที่สุดและเป็นมาตรฐาน (เช่น ตัวเลือกที่ใช้งานมากที่สุด)

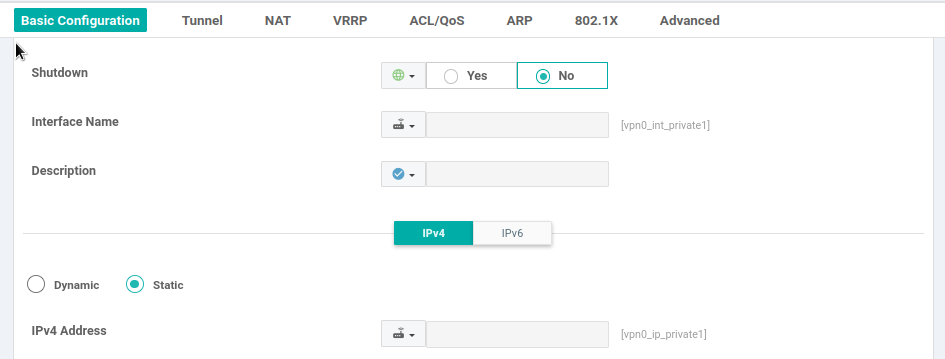
จะมีการแลกเปลี่ยนระหว่างจำนวนเทมเพลตและปริมาณข้อมูลที่ต้องป้อนผ่านตัวแปร เพื่อช่วยในการตัดสินใจนี้ ควรพิจารณาสิ่งที่เป็นไปได้ทั่วไปและสิ่งที่แตกต่างกันในสิ่งที่คุณพยายามทำ ใน topology นี้ ลิงก์ทั้งหมดเป็น full-duplex gigabit Ethernet ที่ใช้ static IP addressing ซึ่งมีลักษณะร่วมกัน ความแตกต่างจะอยู่ที่ชื่อของอินเทอร์เฟซ ว่าจะสามารถรองรับ 802.1Q-tagged traffic หรือไม่ และจะอยู่ใน transport VPN หรือ service VPN

เมื่อเทมเพลตอินเทอร์เฟซถูกกำหนดให้กับ device template แล้ว จะสามารถใช้งานได้เพียงครั้งเดียวต่อ VPN นั่นหมายความว่าสำหรับอุปกรณ์ที่มี dual uplinks จะต้องสร้าง interface templates แยกกันสองชุด เพื่อให้ลดจำนวนเทมเพลต ควรกำหนดสิ่งที่เป็นทั่วไปในจำนวนมากที่สุดของไซต์ของเรา ตัวอย่างเช่น หากไซต์สาขาส่วนใหญ่ถูกตั้งค่าให้ WAN interface แรก เชื่อมต่อกับ MPLS L3VPN ในขณะที่ WAN interface ที่สอง เชื่อมต่อกับ public Internet service เราสามารถสร้างเทมเพลตสำหรับแต่ละประเภทการขนส่ง ซึ่งเป็นสิ่งที่เราจะทำใน lab นี้

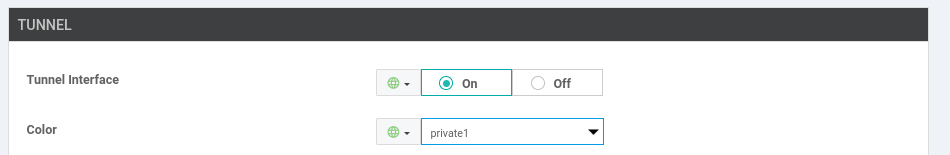
เริ่มต้นโดยการสร้าง feature template ใหม่ แล้วเลือกประเภทอุปกรณ์ CSR1000v และ vEdge Cloud จากนั้นคลิกที่ VPN Interface Ethernet template type ตั้งชื่อเทมเพลตนี้ว่า VPN0\_private1



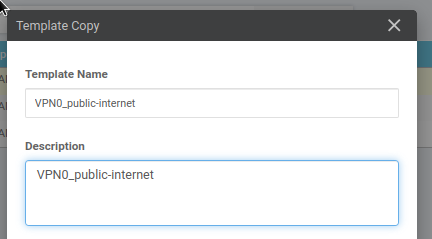
ตั้งค่า Shutdown เป็นค่า global no และตั้งชื่อ Interface Name เป็นตัวแปรเฉพาะอุปกรณ์ vpn0\_int\_private1 นอกจากนี้ยังต้องตั้งค่า IPv4 Address เป็นตัวแปรเฉพาะอุปกรณ์ vpn0\_ip\_private1



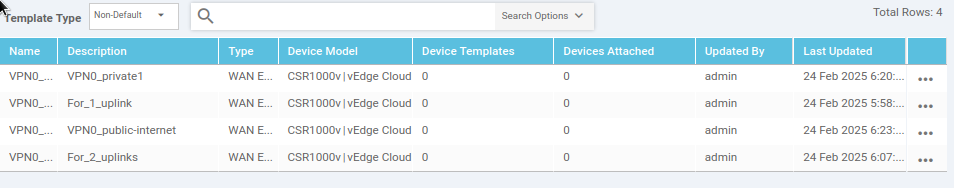
ให้ตั้งค่า Tunnel Interface เป็นค่า global on และ Color เป็นค่า global private1 แล้วคลิก Save ที่ด้านล่างเพื่อทำการบันทึกเทมเพลต



จากนั้น ในส่วนของ Feature Template ให้คัดลอกเทมเพลตก่อนหน้านี้ และตั้งชื่อใหม่ว่า VPN0\_public-internet



ทำการแก้ไขตัวแปรทั้งหมดเพื่อเปลี่ยนจาก private1 เป็น public-internet



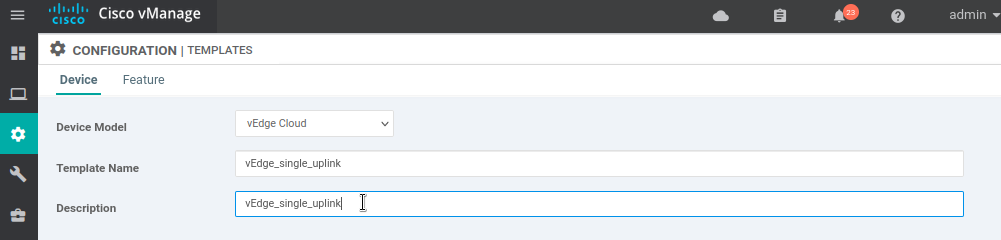
Device templates

เพื่อให้เข้าใจแนวทางการทำงานและเริ่มเห็น templates ในการใช้งาน จะสร้างและนำ device templates ไปใช้สำหรับ SD-WAN edges สามชุดที่เรามีอยู่ Device templates คือชุดรวมของ feature templates นอกจากนี้ยังสามารถสร้าง device templates โดยใช้ CLI ซึ่งอาจเป็นประโยชน์สำหรับการเปลี่ยนไปใช้ SD-WAN จากทักษะในอดีต แต่ถือว่าเป็นวิธีการที่ไม่นิยมใน Lab นี้

การสร้าง device templates จะทำเป็นรายประเภทอุปกรณ์ เพื่อให้สามารถกำหนดค่าฟีเจอร์ที่เกี่ยวข้องกับอุปกรณ์นั้นได้ แต่สามารถมีเทมเพลตหลายชุดสำหรับอุปกรณ์ชนิดเดียวกัน ใน lab นี้เราจะสร้าง device templates สองชุด หนึ่งชุดสำหรับ vEdge และอีกหนึ่งชุดสำหรับ cEdges

สำหรับ lab topology นี้ เราต้องการเพียงสอง device templates เพราะ cEdge sites ทั้งสองแห่งมี dual uplinks หากไซต์หนึ่งมี single uplink ขณะที่อีกไซต์มี dual uplinks เราจะต้องสร้าง device templates แยกกันเพื่อรองรับความแตกต่างนี้ แม้ว่าจะเป็นอุปกรณ์ประเภทเดียวกัน นี่เป็นอีกจุดที่การวางแผนและการมาตรฐานเมื่อต้องนำ Cisco SD-WAN ไปใช้งาน

จากเมนู Configuration > Templates > Device ให้คลิกที่ปุ่ม Create Template และเลือก From Feature Template จากนั้นเลือก vEdge Cloud เป็นโมเดลอุปกรณ์ ตั้งชื่อเทมเพลตนี้ว่า vEdge\_single\_uplink



สำหรับตอนนี้ การตั้งค่าเพียงอย่างเดียวที่เราต้องเปลี่ยนคือการเปลี่ยน VPN 0 template เป็น VPN0\_1\_uplink และ VPN Interface ที่เกี่ยวข้องเป็น VPN0\_private1



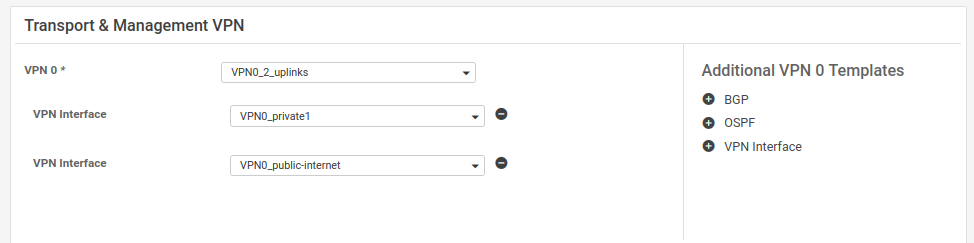
แม้ว่าเราจะไม่ใช้ VPN 512 interface ในขณะนี้ แต่ยังคงต้องระบุในเทมเพลต สำหรับ vEdge ดังนั้นให้เลือก factory default, VPN interface และเพิ่ม default template จากนั้นคลิกปุ่ม Create ที่ด้านล่าง



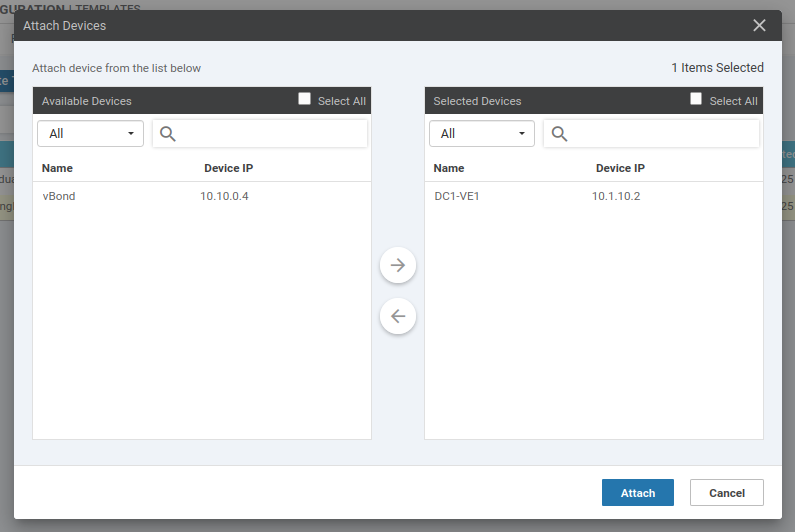
สร้าง device template อีกชุดหนึ่ง และเลือก CSR1000v โดยตั้งชื่อว่า CSR1Kv\_dual\_uplink ในเวอร์ชันของ vManage code นี้ cEdge templates จะระบุ AAA และ Cisco-AAA templates ซึ่งไม่สามารถใช้ร่วมกันได้ ให้เปลี่ยน AAA template เป็น None ตั้งค่า VPN 0 เป็น VPN0\_2\_uplinks และ VPN Interface เป็น VPN0\_private1



ในส่วน Additional VPN 0 Templates ให้เพิ่ม VPN Interface ใหม่และตั้งค่าเป็น VPN0\_public-internet จากนั้นคลิกปุ่ม Create ที่ด้านล่าง

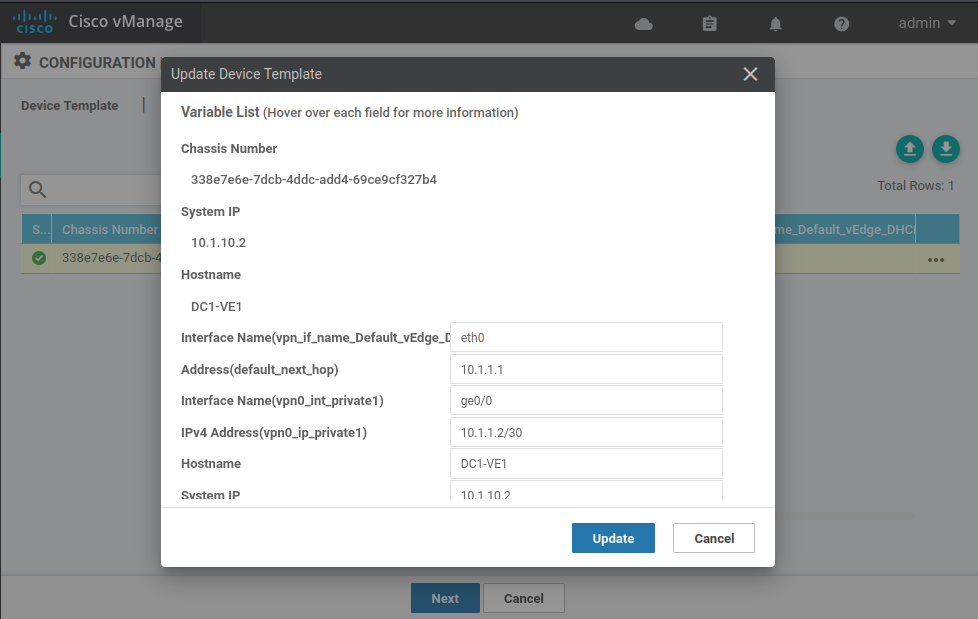


ตอนนี้เราจะนำเทมเพลตไปใช้กับอุปกรณ์ ให้คลิกที่สามจุดด้านขวาของแถว vEdge\_single\_uplink และเลือก Attach Devices จากนั้นเลือก DC1-VE1 จากรายการแล้วคลิก Attach



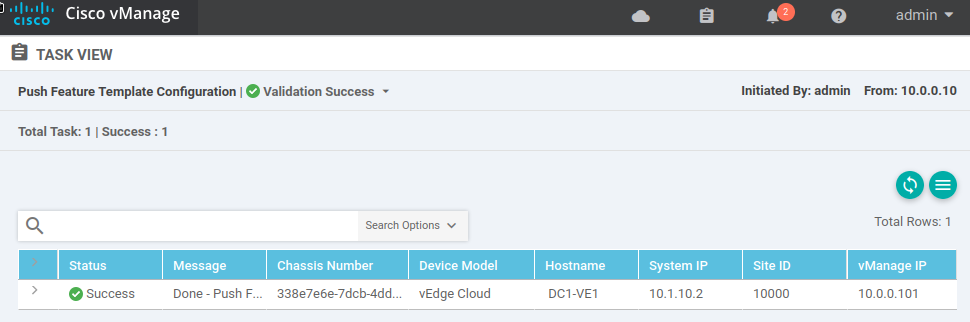
ในขั้นตอนนี้ เรามีตัวเลือกในการกำหนดค่าตัวแปรแต่ละตัวโดยตรง หรือการนำเข้าและส่งออกไฟล์ CSV หากเรามีอุปกรณ์มากกว่าห้าชุด การกำหนดค่าอุปกรณ์โดยใช้ไฟล์ CSV จะรวดเร็วกว่ามาก

คลิกที่สามจุดแล้วเลือก Edit Device Template ตอนนี้ เราจะเห็นตัวแปรที่เราสร้างขึ้นภายใน feature templates เราควรจะสามารถเห็นได้ทันทีว่าการตั้งชื่อตัวแปรที่ดีและมีความหมายมีความสำคัญมาก ตัวแปรและค่าของ DC1-VE1 ได้แก่:



คลิก Update แล้วคลิก Next

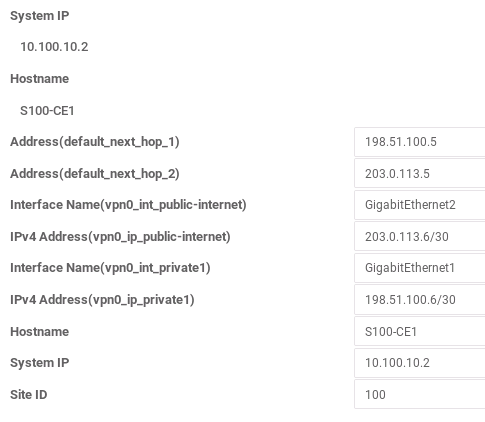
ตอนนี้เราจะเห็นหน้าจอ ‘pre-provisioning’ ไม่ต้องทำอะไรที่นี่ และสามารถคลิก Configure Devices เพื่อดำเนินการต่อหากต้องการ อย่างไรก็ตาม เราสามารถคลิกที่อุปกรณ์ใดก็ได้ในรายการเพื่อดูว่าการกำหนดค่าที่จะนำไปใช้เป็นอย่างไร เปรียบเทียบกับการกำหนดค่าปัจจุบัน และตั้งค่าตัวจับเวลาการย้อนกลับในกรณีที่การกำหนดค่าใหม่ทำให้การสื่อสารกับ controller ล้มเหลว โดยเฉพาะฟีเจอร์ config diff นั้นมีประโยชน์มาก หลังจากที่เราคลิก Configure Devices การกำหนดค่าจะถูกส่งไปยังอุปกรณ์ต่างๆ เราสามารถดูความก้าวหน้า และหากเกิดข้อผิดพลาด การกำหนดค่าจะถูกย้อนกลับและคุณควรได้รับข้อมูลเกี่ยวกับสาเหตุที่การกำหนดค่าล้มเหลว



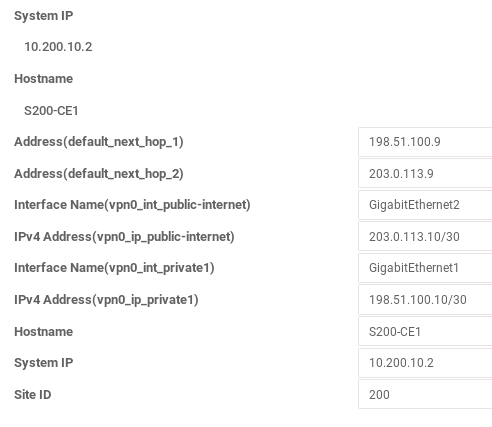
กลับไปที่ส่วน device templates และแนบ cEdges สองตัวเข้ากับ CSR1Kv template นี่คือค่าที่ผู้สอนใช้สำหรับ

lab นี้:

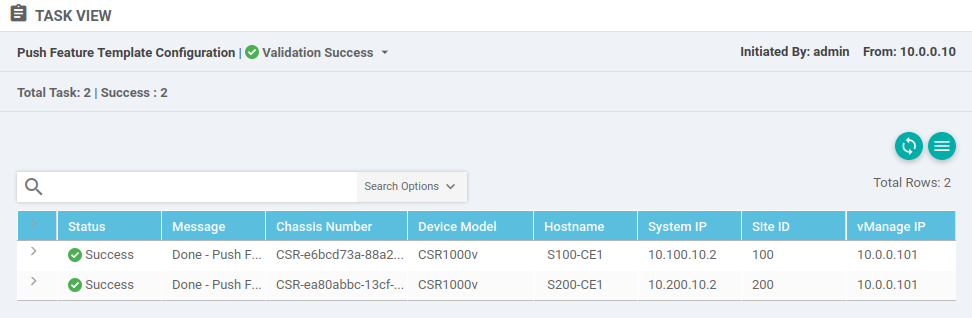
**S100-CE1:**



**S200-CE1:**



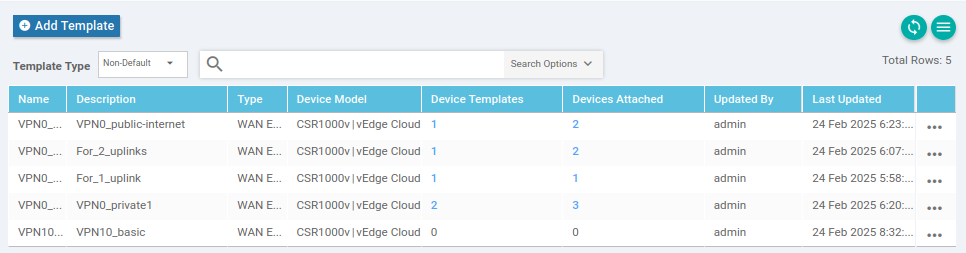
เมื่อค่าทั้งหมดถูกป้อนและการกำหนดค่าได้ถูกนำไปใช้ หวังว่าทุกอย่างจะเป็นไปด้วยดีและอุปกรณ์ทั้งหมดของเราจะออนไลน์อยู่ ณ จุดนี้ ตอนนี้เราพร้อมที่จะเริ่มนำข้อมูลผู้ใช้เข้าสู่วิถี SD-WAN แล้ว



Service VPN

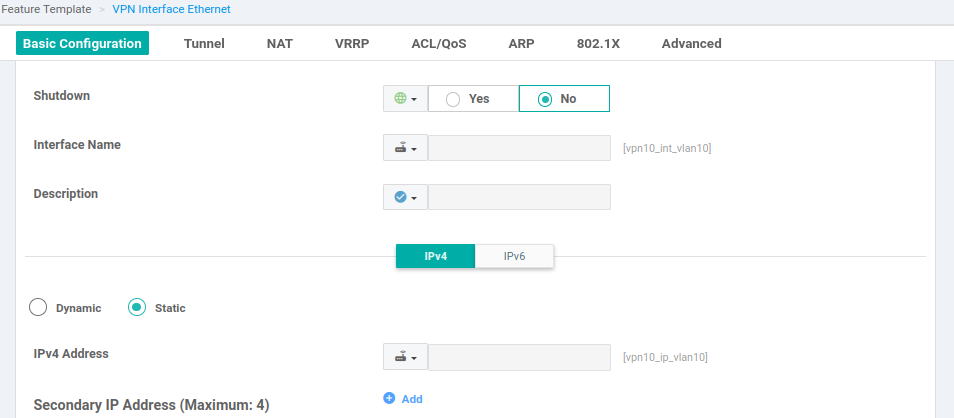
อย่างที่เราได้เห็นแล้ว จะใช้เวลาค่อนข้างมากในการเตรียมความพร้อมก่อนที่คุณจะเริ่มส่งข้อมูลผู้ใช้ผ่าน SD-WAN ได้ โชคดีที่กระบวนการพื้นฐานในการเพิ่ม Service VPN นั้นมีความคล้ายคลึงกับสิ่งที่เราได้กล่าวถึงไปแล้ว Service VPNs ใช้ในการส่งข้อมูลผู้ใช้ และเป็น VPN ใด ๆ ที่มี ID แตกต่างจาก 0 หรือ 512 ตั้งแต่เวอร์ชัน v19.2 เป็นต้นไป เราสามารถมี VPN ทั้งหมด 64 รายการใน fabric เดียว

จากส่วน configuration template ใน vManage ให้สร้าง feature template ใหม่และเลือกอุปกรณ์ CSR1000V และ vEdge Cloud จากนั้นเลือก VPN template สำหรับ template นี้ ตั้งชื่อว่า VPN10\_basic ตั้งค่า VPN global 10 ในส่วน Advertise OMP ให้ตั้งค่า Static และ Connected เป็น global on แล้วคลิกที่ปุ่ม Save ที่ด้านล่าง

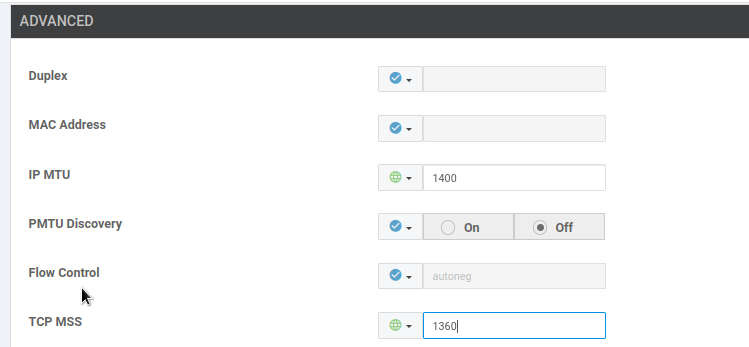


ถัดไป เราต้องสร้าง VPN interface templates สำหรับ Service VPN นี้ อีกครั้ง เราต้องพิจารณาว่าสำหรับแต่ละอุปกรณ์ เราสามารถใช้ VPN interface template ได้เพียงครั้งเดียวต่อ VPN ดังนั้นหากเรามีหลาย interface ที่เข้าร่วมใน VPN เดียวกันบนอุปกรณ์เดียว เราต้องใช้ interface template ที่แตกต่างกันสำหรับแต่ละ interface สำหรับ lab นี้ อุปกรณ์ SD-WAN ทั้งหมดของเรามี LAN-facing subinterfaces สามตัว ดังนั้นเราจึงต้องการ VPN interface templates สามตัว

ให้สร้าง feature template ใหม่ โดยเลือกอุปกรณ์ CSR1000V และ vEdge Cloud และเลือก VPN Interface Ethernet template ตั้งชื่อว่า VPN10\_int\_vlan10 ตั้งค่าค่า Shutdown เป็น global no และตั้งชื่อ interface เป็นตัวแปรเฉพาะของอุปกรณ์ vpn10\_int\_vlan10 เรายังใช้ที่อยู่ IP แบบ static ดังนั้น ให้ตั้งค่าที่อยู่ IPv4 เป็นตัวแปรเฉพาะของอุปกรณ์ vpn10\_ip\_vlan10

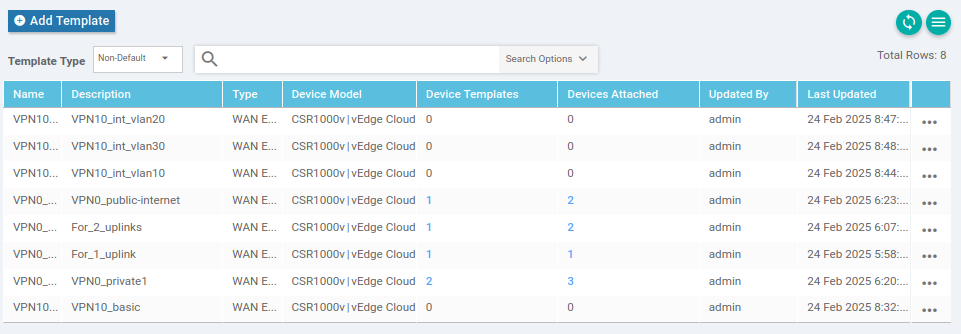


การตั้งค่าครั้งสุดท้ายที่เราจะเปลี่ยนแปลงสำหรับ template นี้ในขณะนี้คือภายใต้ส่วน Advanced เราต้องเปลี่ยนค่า IP MTU เป็น global 1400 และ TCP MSS เป็น global 1360 นี่คือค่าที่เป็น ‘catch-all’ และก็ใช้ได้ดีสำหรับ lab แต่เป็นสิ่งที่ต้องพิจารณาอย่างจริงจังในเครือข่าย Production ด้วย DMVPN tunnels การตั้งค่าเหล่านี้เป็นสิ่งที่พบได้ทั่วไป มาตรฐาน MTU คือ 1500 และอุปกรณ์ edge จะต้องทำการแบ่งส่วน packet หาก MTU ถูกตั้งค่าต่ำกว่าและอุปกรณ์ที่เชื่อมต่อกับ subnet ไม่ได้รับการปรับเปลี่ยนให้สอดคล้อง



ในทางปฏิบัติ สำหรับข้อมูลผู้ใช้ทั่วไป นี่อาจไม่ใช่ปัญหาใหญ่ โดยเฉพาะสำหรับสำนักงานสาขาขนาดเล็กที่มีปริมาณการจราจรต่ำ อีกสิ่งที่ต้องพิจารณาในออกแบบ Cisco SD-WAN คือ หากgikจะใช้ 802.1Q-tagged subinterfaces (ซึ่งเราจะใช้ใน lab นี้) เมื่อตั้งค่า subinterfaces ที่ติดแท็ก เราจะต้องตั้งค่า parent interface ให้มี MTU ที่สูงกว่าอย่างน้อย 4 ไบต์ (เช่น 1504) หรือกำหนดให้ subinterfaces ทั้งหมดมี MTU ต่ำกว่าที่ตั้งไว้ 4 ไบต์ เมื่อเราตั้งค่า MTU เป็น 1400 และตรวจสอบว่าอุปกรณ์ที่เชื่อมต่อของเรายังได้รับการกำหนดค่าให้สอดคล้อง เราจะสามารถรองรับความต้องการเพิ่มเติมจากการ encapsulations ต่างๆ (802.1Q, IPsec, ฯลฯ) ได้

ให้คัดลอก Template อีกสองครั้งและเปลี่ยนค่าของ VLAN เป็น 20 และ 30 ตามลำดับ

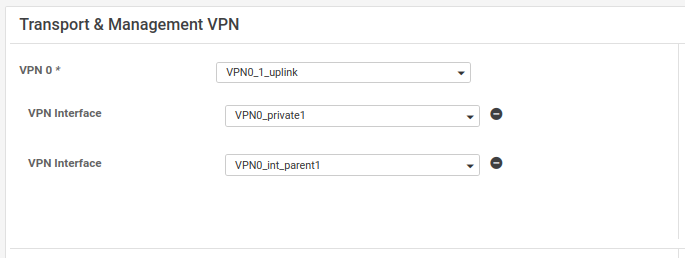


เมื่อใช้ subinterfaces, parent interface จะต้องมี template ของตัวเองด้วย และมันจะต้องอยู่ใน VPN 0 หากเรามี subinterfaces โดยไม่มี parent interface ที่อยู่ใน VPN 0 เราจะได้รับข้อผิดพลาดเมื่อพยายามส่ง template ออกไป ถึงแม้ว่า parent interface จะอยู่ใน VPN 0 แต่มันจะไม่ทำการเชื่อมต่อ tunnels และจะไม่มีการกำหนด IP address ซึ่งจะทำหน้าที่เป็น placeholder

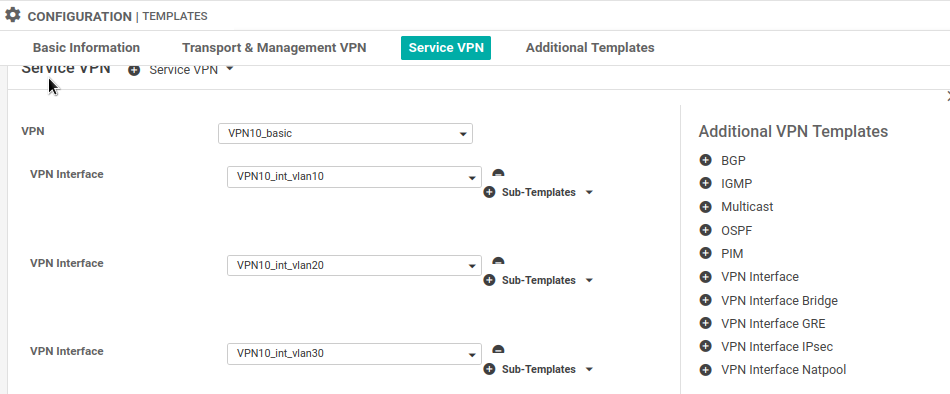
เราสามารถคัดลอกหนึ่งใน template ก่อนหน้านี้และเปลี่ยนชื่อเป็น VPN0\_int\_parent1 เปลี่ยนชื่อ Interface เป็น vpn0\_int\_parent1 และตั้งค่าที่อยู่ IPv4 เป็นค่า default นอกจากนี้ให้ตั้งค่า IP MTU และ TCP MSS เป็นค่า default ด้วย



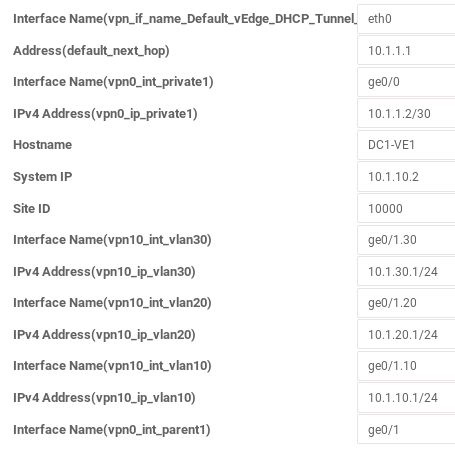
ตอนนี้เราสามารถแนบ template ไปยังอุปกรณ์ได้ จากส่วนการ configuration device templates ให้แก้ไข template ของ vEdge ภายใต้ VPN 0 เพิ่ม VPN Interface ใหม่และอ้างอิงถึง template VPN0\_int\_parent1

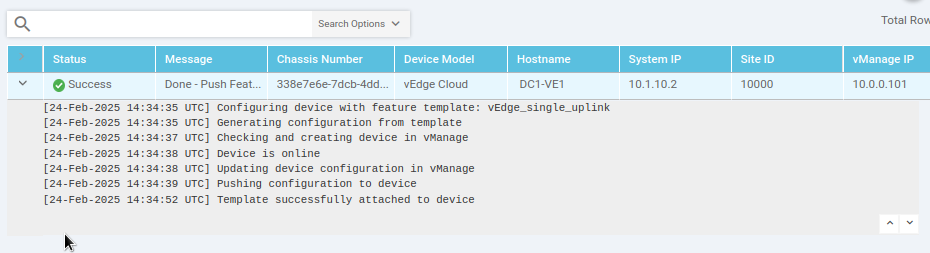


จากนั้นคลิกที่เครื่องหมายบวกถัดจาก Service VPN และเลือก VPN10\_basic จาก dropdown คลิก VPN Interface สามครั้งและเลือกสาม VLAN interface templates จากนั้นคลิกที่ปุ่ม Update เพื่อบันทึกการเปลี่ยนแปลงของ device template



เราจะเห็นรายการอุปกรณ์ที่ต้องกรอกค่าตัวแปร คลิกที่สามจุดทางด้านขวาของอุปกรณ์และเลือก Edit Device Template ค่าที่ใช้สำหรับ lab นี้มีดังนี้:





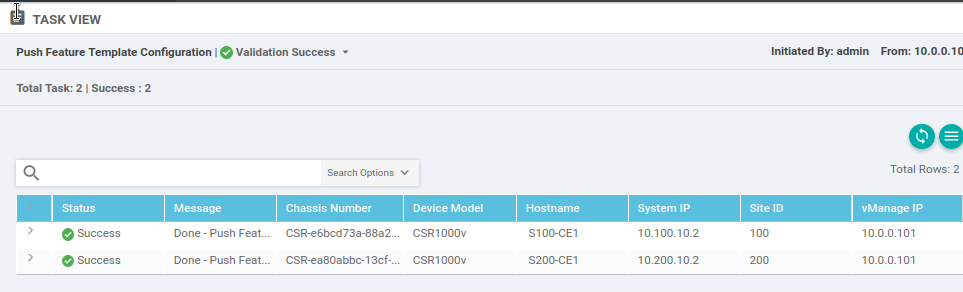
ทำตามขั้นตอนเดียวกันสำหรับเทมเพลตอุปกรณ์ cEdge เพิ่มอินเทอร์เฟซหลัก VPN 0, VPN 10 service VPN และสามซับอินเทอร์เฟซ

S100-CE1:

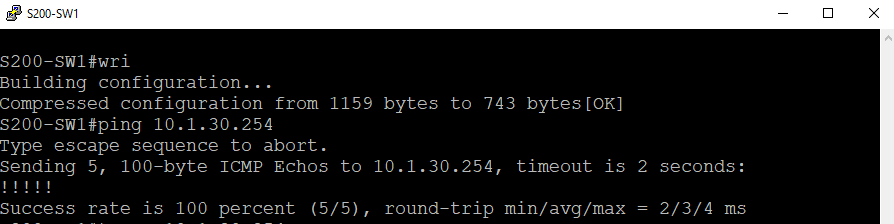


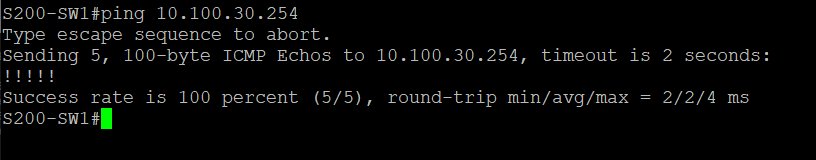
S200-CE1:



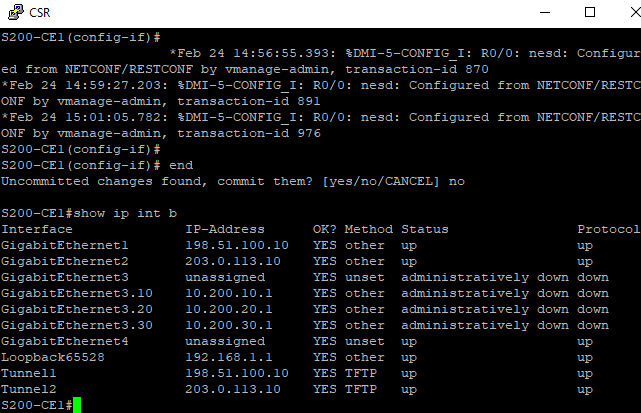


สำเร็จ!   
ในขั้นตอนนี้ เราควรสามารถส่ง pings ไปยังที่อยู่ .1 หรือ .254 ใดๆ จากทั้งเราเตอร์ edge และสวิตช์ที่เชื่อมต่ออยู่ภายใน VPN 10 ได้

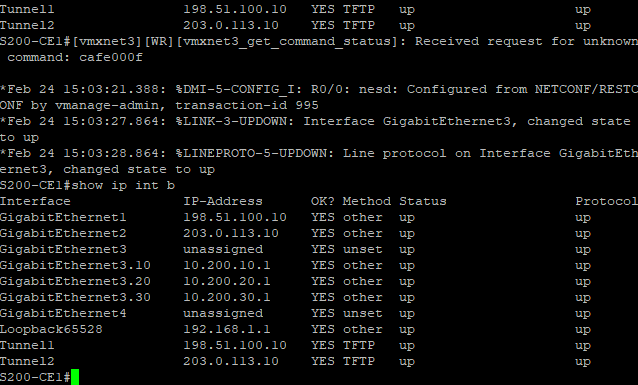


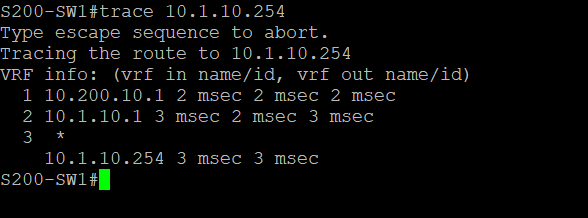


ในกรณีที่ Ping ไม่ได้ ลองตรวจสอบการกำหนดค่าใน Templates อีกรอบ ว่าในส่วนของ Shutdown ได้เลือกเป็น No หรือไม่ เพราะเราอาจจะไม่ได้สั่งเปิด Port ผ่าน Template



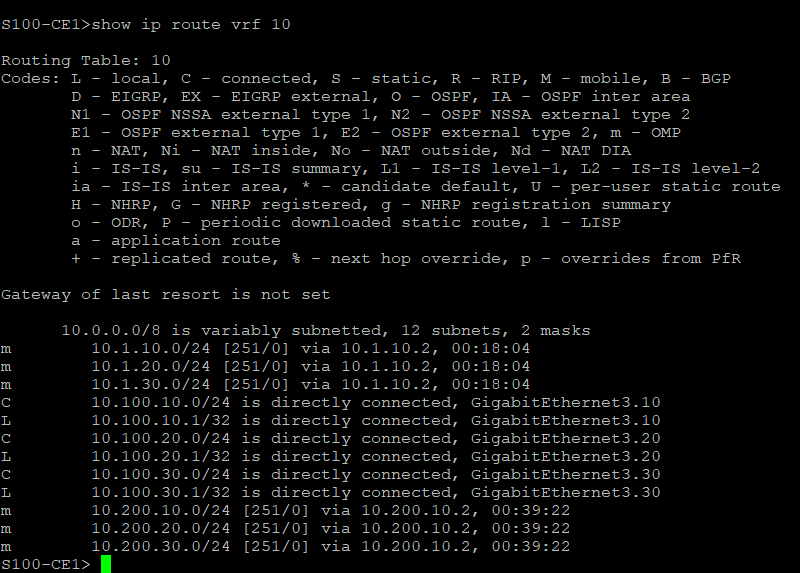
แก้ไขแล้ว Push ใหม่



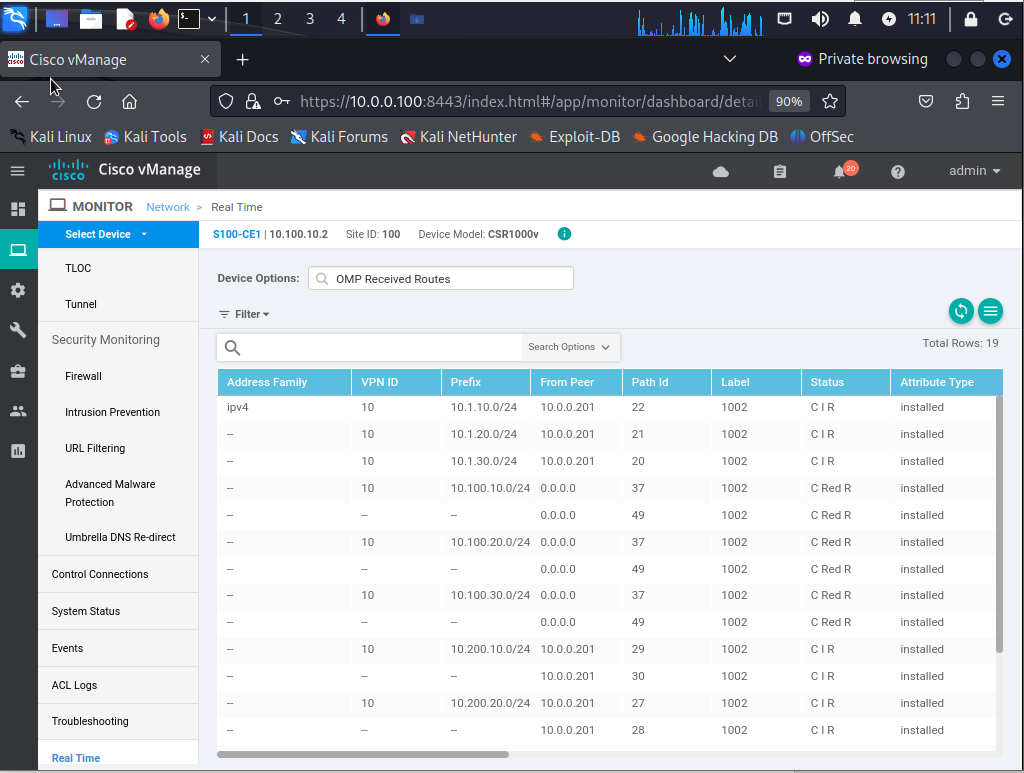


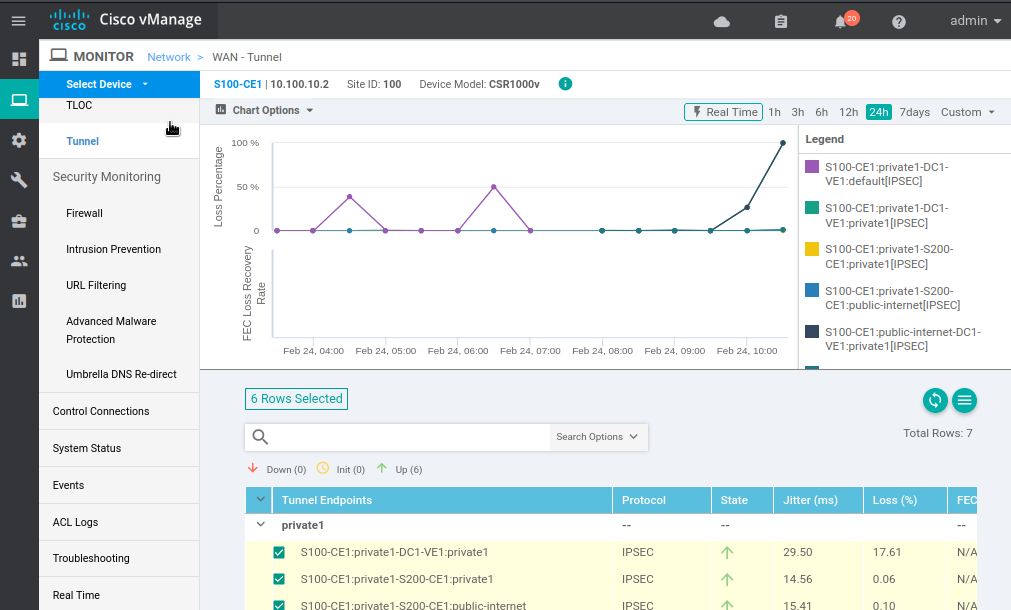
การใช้คำสั่ง traceroute จะแสดงให้เห็นว่า S200-CE1 เป็น hop แรก จากนั้นการขนส่งพื้นฐานจะมองไม่เห็นโดยสิ้นเชิงจนกว่าจะถึง DC1-VE1 (เหมือนกับ MPLS L3VPN ที่ปิดการเผยแพร่ TTL) จริงๆ แล้วสิ่งนี้ลึกกว่าการใช้งาน L3VPN ของ SP แบบดั้งเดิม เพราะว่า edge จะอยู่หลังการกระโดดหลายครั้งภายในศูนย์ข้อมูลที่อยู่หลังไฟร์วอลล์ edge ที่เชื่อมต่อกับ SP

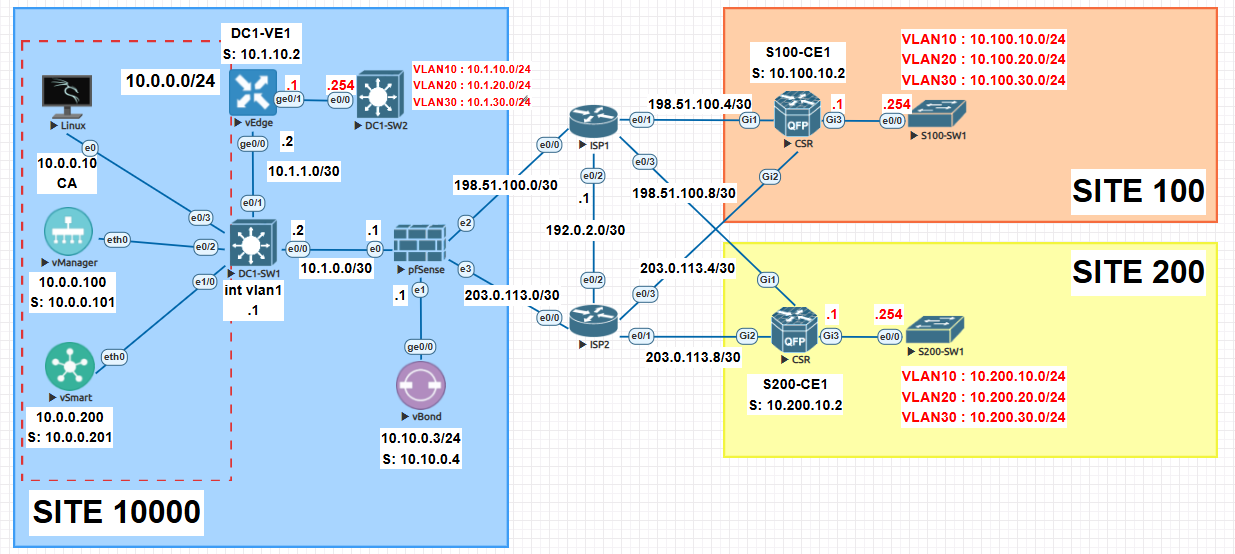
จาก cEdge เราสามารถดูตารางการกำหนดเส้นทาง VPN ได้เช่นเดียวกับ VRF แบบดั้งเดิม:



เราสามารถดูรายละเอียดเพิ่มเติมจากบรรทัดคำสั่งด้วยคำสั่งต่างๆ เช่น show sdwan omp แต่แน่นอนว่านั่นเป็นวิธีการแบบเก่า เราสามารถเข้าถึงข้อมูลเดียวกัน (และข้อมูลเพิ่มเติมอีกมาก) ผ่าน vManage โดยไปที่ Monitor > Network > Device > Real Time และเลือกข้อมูลที่ต้องการ เช่น OMP Received Routes







#### Set up the CA

**Root Certificate Authority (CA)** สำหรับ Cisco SD-WAN ที่ใช้ **X.509 certificates** เป็นกลไกความเชื่อถือ (Trust Model) อุปกรณ์ใด ๆ ใน SD-WAN fabric จะต้องมีใบรับรองที่ถูกต้องติดตั้งไว้ มิฉะนั้นจะไม่สามารถเข้าร่วมเครือข่ายได้

sudo -i

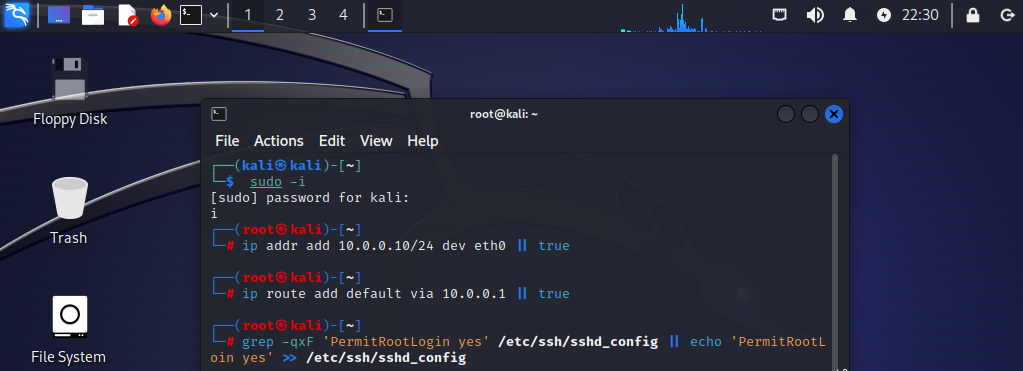
ip addr add 10.0.0.10/24 dev eth0 || true

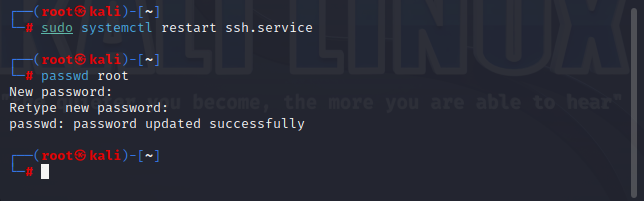
ip route add default via 10.0.0.1 || true

grep -qxF 'PermitRootLogin yes' /etc/ssh/sshd\_config || echo 'PermitRootLogin yes' >> /etc/ssh/sshd\_config

sudo systemctl restart ssh.service

passwd root (ตั้งรหัสผ่าน)

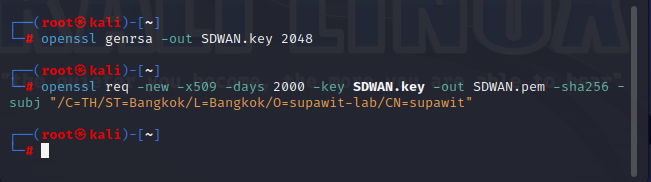




openssl genrsa -out SDWAN.key 2048

openssl req -new -x509 -days 2000 -key SDWAN.key -out SDWAN.pem -sha256 \

-subj "/C=TH/ST=Bangkok/L=Bangkok/O=supawit-lab/CN=supawit"



#### Set up the DC1-SW1

hostname DC1-SW1

no ip domain-lookup

!

!

interface Ethernet0/0

no switchport

ip address 10.1.0.2 255.255.255.252

!

interface Ethernet0/1

no switchport

ip address 10.1.1.1 255.255.255.252

!

interface Vlan1

ip address 10.0.0.1 255.255.255.0

no shut

!

ip forward-protocol nd

!

ip http server

ip http secure-server

!

ip route 0.0.0.0 0.0.0.0 10.1.0.1

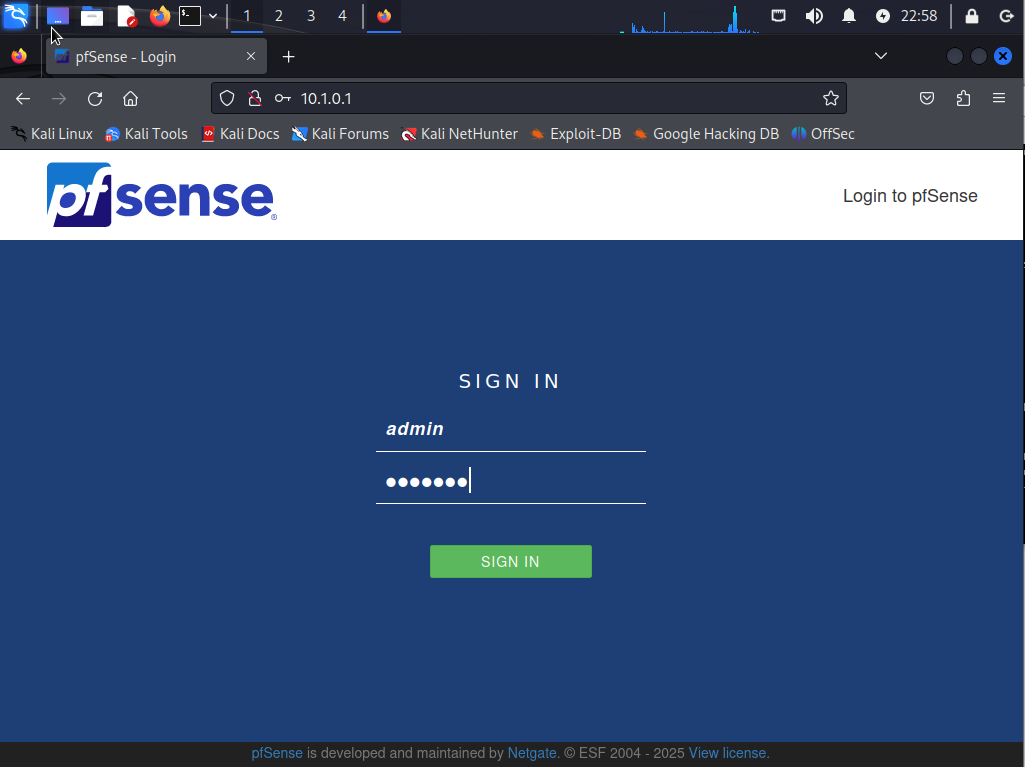
!

!

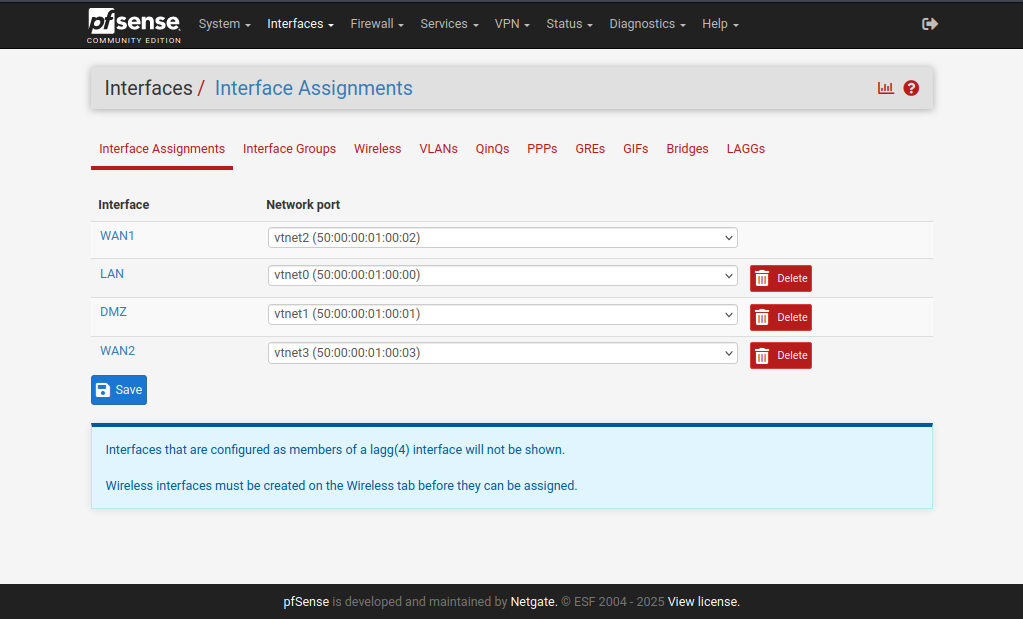
end

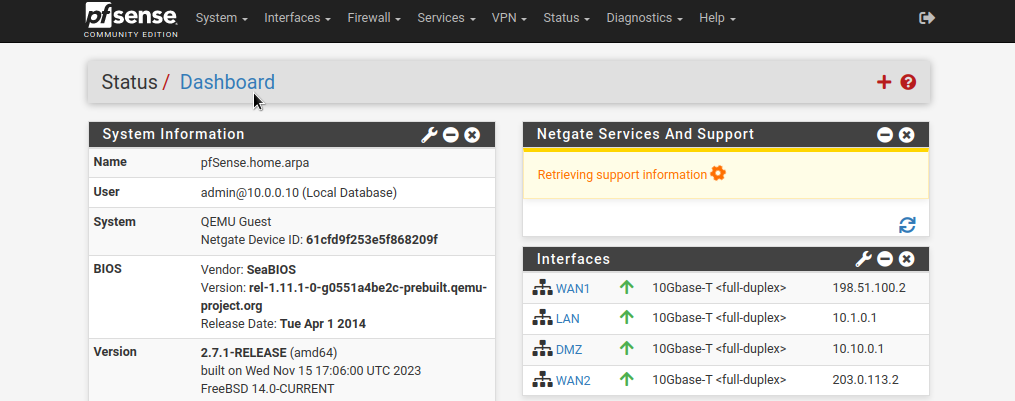
#### Set up the Firewall

เนื่องจากในแล็บนี้เราโฟกัสที่ SDN จึงไม่ได้สอน setup firewall แบบละเอียด หรือเราจะใช้ Router มาแทนหน้าที่ได้ (admin:pfsense)

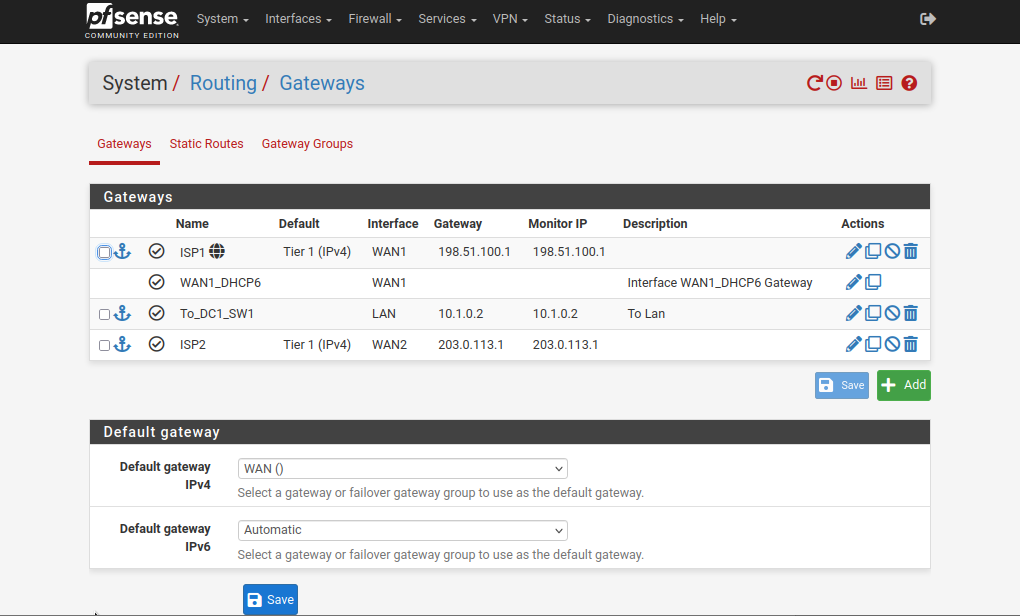


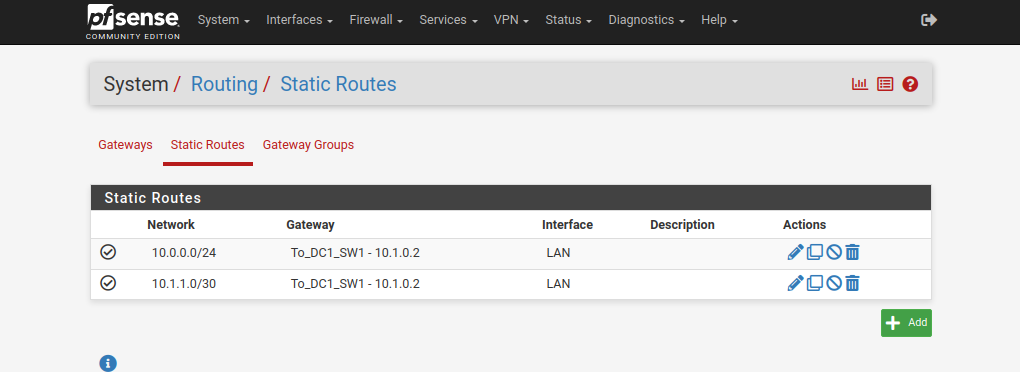
**Interfaces Assignment**

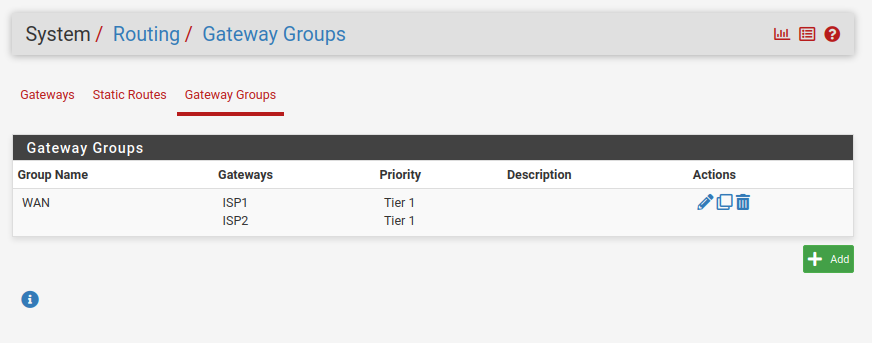




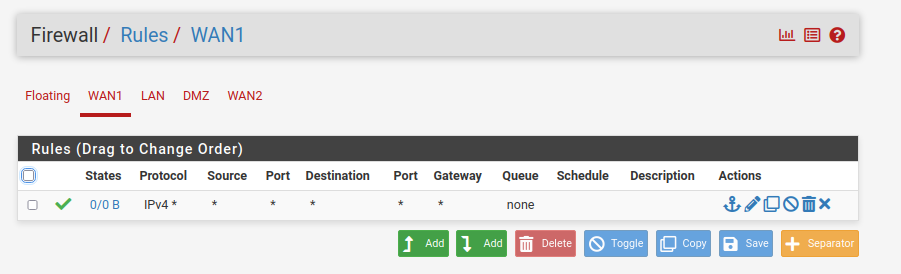
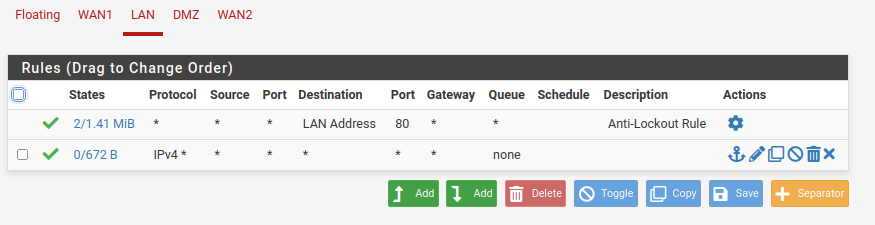
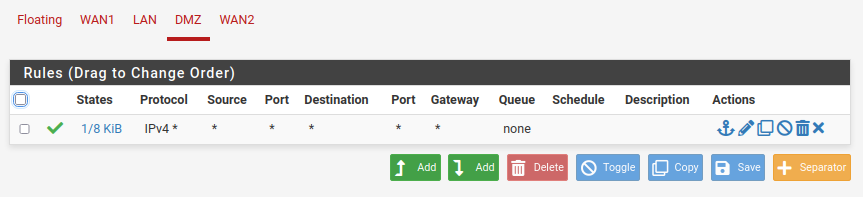
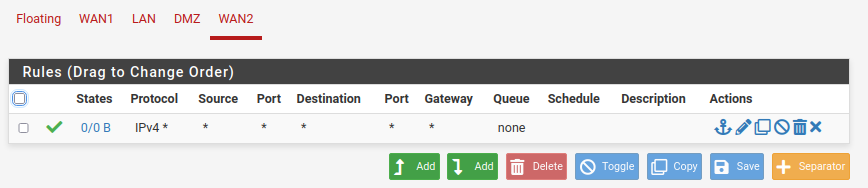
**Routing Config**







**Policy Config** (ในที่นี้ allow ทุกอย่าง เพื่อให้ง่ายในการศึกษาแล็บนี้)

#### Set up the ISP1

hostname ISP1

no ip domain lookup

!

interface Ethernet0/0

ip address 198.51.100.1 255.255.255.252

no shut

interface Ethernet0/1

ip address 198.51.100.5 255.255.255.252

no shut

!

interface Ethernet0/2

ip address 192.0.2.1 255.255.255.252

no shut

!

interface Ethernet0/3

ip address 198.51.100.9 255.255.255.252

no shut

!

router ospf 1

network 0.0.0.0 255.255.255.255 area 0

!

!

ip route 10.0.0.0 255.0.0.0 198.51.100.2

ip route 10.0.0.0 255.0.0.0 192.0.2.2 5

!

end

#### Set up the ISP2

hostname ISP2

no ip domain lookup

!

interface Ethernet0/0

ip address 203.0.113.1 255.255.255.252

no shut

!

interface Ethernet0/1

ip address 203.0.113.9 255.255.255.252

no shut

!

interface Ethernet0/2

ip address 192.0.2.2 255.255.255.252

no shut

!

interface Ethernet0/3

ip address 203.0.113.5 255.255.255.252

no shut

!

router ospf 1

network 0.0.0.0 255.255.255.255 area 0

!

ip forward-protocol nd

!

!

ip route 10.0.0.0 255.0.0.0 203.0.113.2

ip route 10.0.0.0 255.0.0.0 192.0.2.1 5

!

end

#### Set up the vBond

conf t

system

host-name vBond

system-ip 10.10.0.4

site-id 10000

organization-name supawit-lab

vbond 10.10.0.3 local vbond-only

!

vpn 0

interface ge0/0

ip address 10.10.0.3/24

no tunnel-interface

no shutdown

ip route 0.0.0.0/0 10.10.0.1

!

commit and-quit

vshell

scp root@10.0.0.10:SDWAN.pem .

exit

request root-cert-chain install /home/admin/SDWAN.pem

* ค่าที่ตั้งใน organization-name ต้องตรงกับที่ใช้ใน PnP vBond profile และต้องเหมือนกันทุกอุปกรณ์ใน SD-WAN Fabric
* อุปกรณ์ทุกตัวใน Fabric ต้องรู้วิธีเข้าถึง vBond
* ไฟล์ vBond image ใช้ไฟล์เดียวกันกับ vEdge Cloud แต่การตั้งค่า vbond-only บน VM จะทำให้ระบบรู้ว่าอุปกรณ์นี้ทำหน้าที่เป็น vBond

ความหมายของ VPN ใน Cisco SD-WAN

* VPN ใน Cisco SD-WAN คล้ายกับ VRF (Virtual Routing and Forwarding)
* VPN0 คือ Transport VPN ที่ใช้กับทุกอุปกรณ์ใน SD-WAN
* Static Route ที่กำหนด ใช้ได้เฉพาะใน VPN0
* IPsec / GRE Tunnels จะถูกสร้างขึ้นบน VPN0
* การตั้งค่า ‘skinny’ configuration ปิดการทำงานของ VPN0 Tunnel Interface ชั่วคราว จนกว่าจะตั้งค่า vSmart และ vManage เสร็จ เพื่อหลีกเลี่ยงปัญหา “ไก่กับไข่” (ปัญหาการเชื่อมต่อและความเชื่อถือของ Certificate)



#### Set up the v Smart

conf t

system

host-name vSmart

system-ip 10.0.0.201

site-id 10000

organization-name supawit-lab

vbond 10.10.0.3

!

vpn 0

interface eth0

ip address 10.0.0.200/24

no tunnel-interface

no shutdown

ip route 0.0.0.0/0 10.0.0.1

!

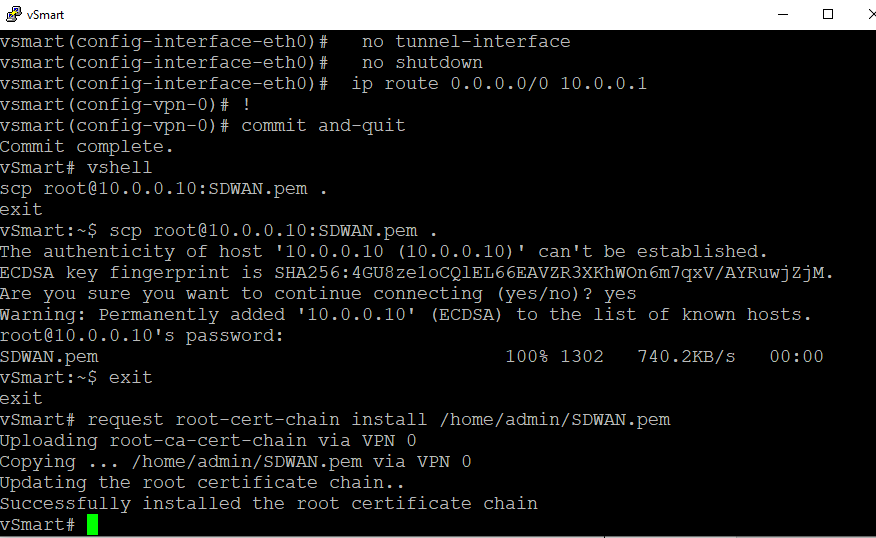
commit and-quit

vshell

scp root@10.0.0.10:SDWAN.pem .

exit

request root-cert-chain install /home/admin/SDWAN.pem



#### Set up the vManage

conf t

system

host-name vManage

system-ip 10.0.0.101

site-id 10000

organization-name supawit-lab

vbond 10.10.0.3

!

vpn 0

interface eth0

ip address 10.0.0.100/24

no shutdown

ip route 0.0.0.0/0 10.0.0.1

!

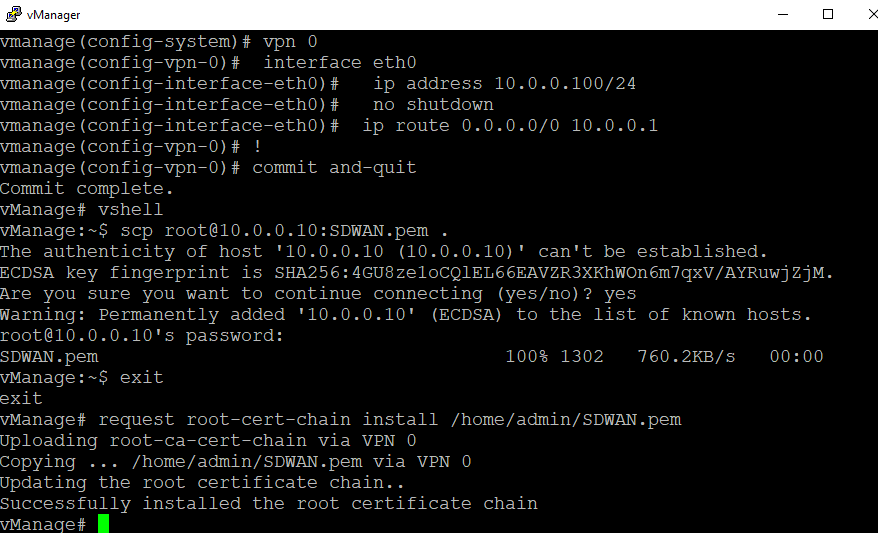
commit and-quit

vshell

scp root@10.0.0.10:SDWAN.pem .

exit

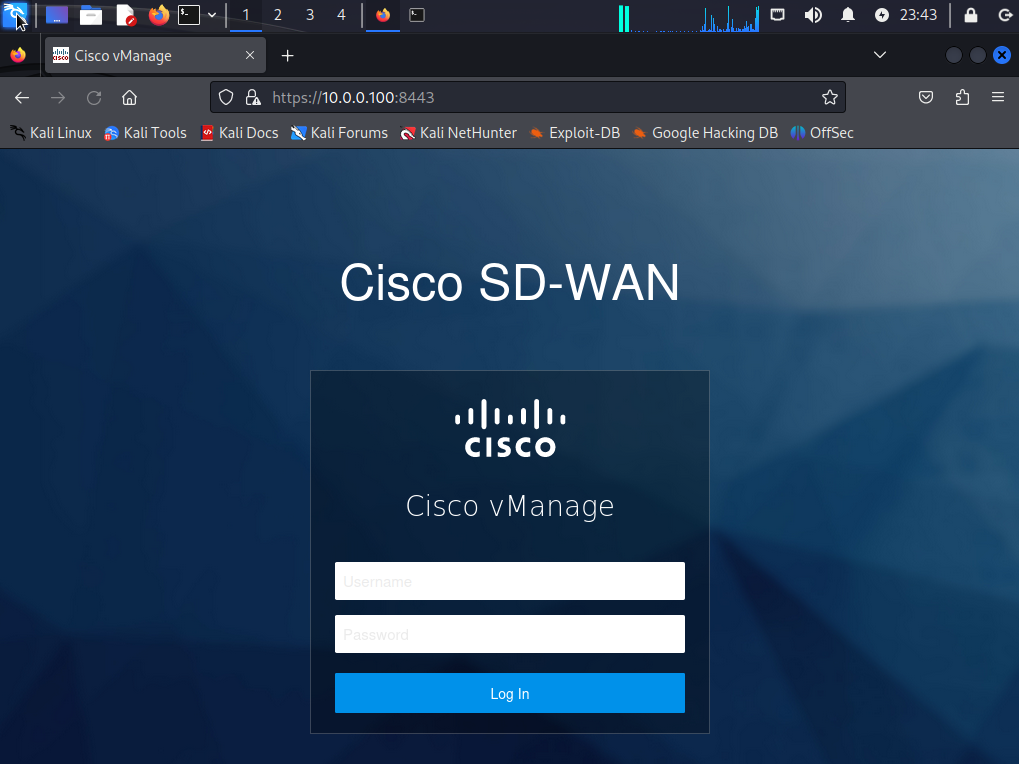
request root-cert-chain install /home/admin/SDWAN.pem



ตอนนี้ให้ใช้ เว็บเบราว์เซอร์ (จากเครื่อง CA ในแล็บนี้ และสำหรับตัวอย่างทั้งหมดต่อจากนี้) เพื่อเข้าสู่ vManage Web Console ที่

https://10.0.0.100:8443

หมายเหตุ: หาก VM เพิ่งรีบูต อาจต้องรอสักครู่ก่อนที่ Web Interface จะพร้อมใช้งาน

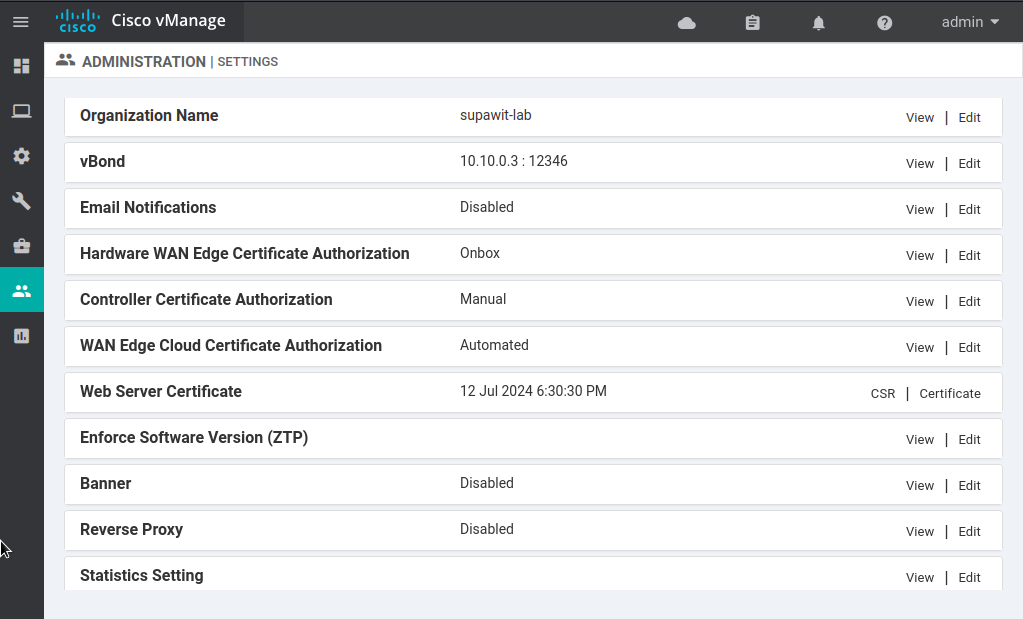


เมื่อเข้าใช้งาน vManage ครั้งแรก ระบบจะแสดง คำเตือนเกี่ยวกับ Self-Signed Certificate ซึ่งสามารถ เพิกเฉย ได้ เนื่องจากเป็นแค่แล็บทดสอบ

เข้าสู่ระบบด้วย: admin / admin

ตั้งค่าบน vManage

1. ไปที่ Administration > Settings
2. แก้ไขค่า Organization Name ให้ตรงกับค่าที่ตั้งไว้ในอุปกรณ์อื่น ๆ (supawit-lab)
3. แก้ไขค่า vBond และระบุ vBond Address
4. ตั้งค่า Controller Certificate Authorization เป็น Manual (เนื่องจากแล็บนี้ใช้ Self-Signed Certificates หากใช้ PKI (Public Key Infrastructure) ขององค์กรที่มี CA Hierarchy สามารถเลือก Enterprise Root Certificate ได้แทน

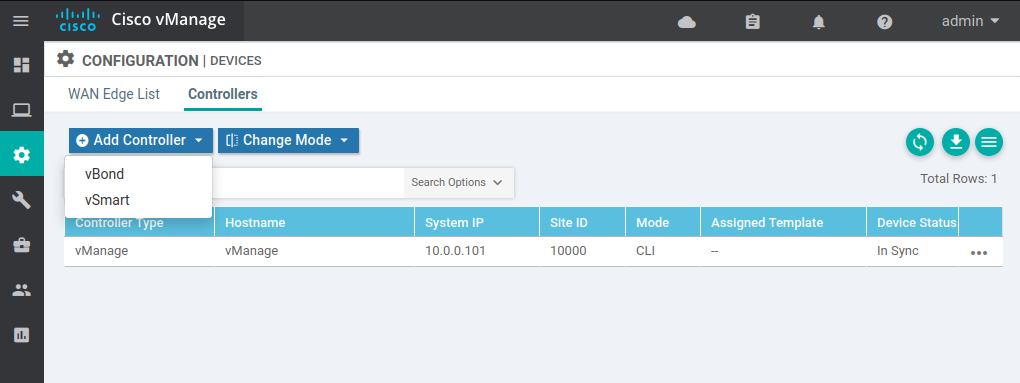


#### Prepare the controller certificates

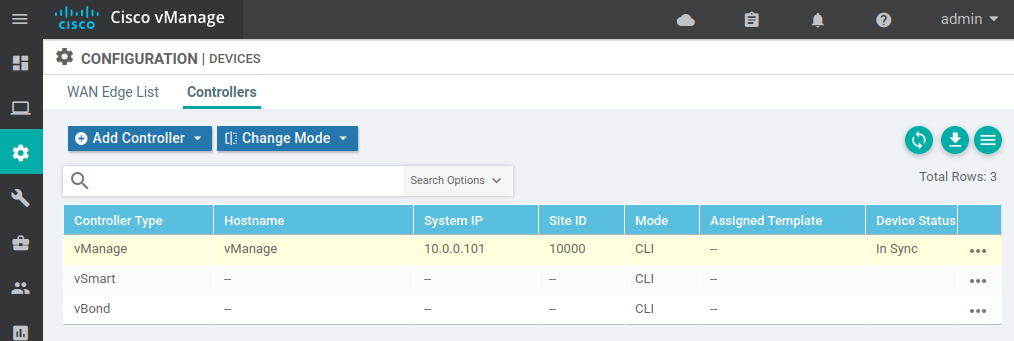
**เพิ่ม vBond และ vSmart บน vManage**

ไปที่ vManage และทำตามขั้นตอนนี้

1. ไปที่ Configuration > Devices > Controllers > Add Controller > vBond

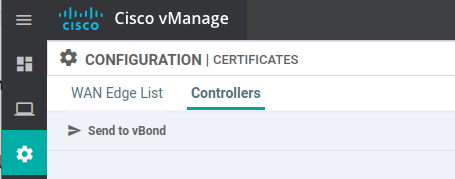


1. ป้อน vBond IP Address ที่ใช้กำหนดไว้ทุกที่ (ในตัวอย่างนี้คือ 10.10.0.3)
2. ป้อน admin username และ password
3. ยกเลิกการเลือก "Generate CSR"
4. ทำตามขั้นตอนเดียวกันเพื่อเพิ่ม vSmart Controller (ใช้ IP 10.0.0.200)
5. ใช้ค่า Protocol เป็นค่าเริ่มต้น (DTLS) และยัง ไม่ต้องสร้าง CSR

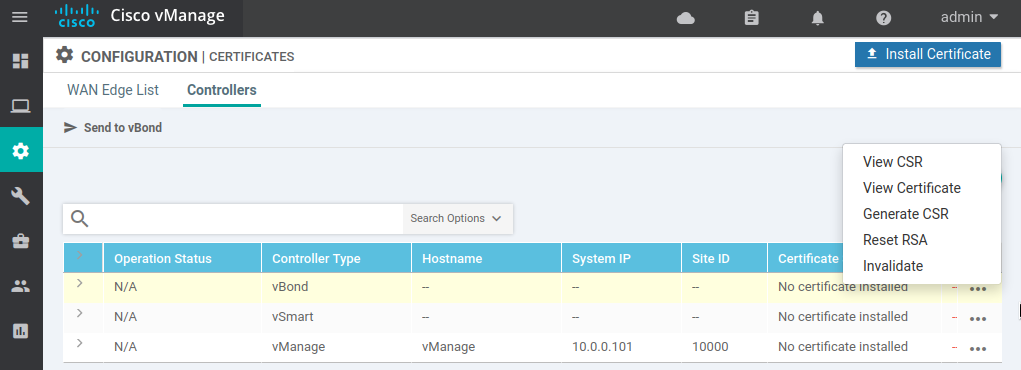


**สร้าง CSR (Certificate Signing Request)**

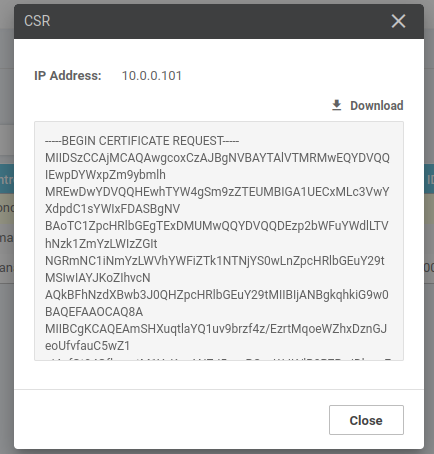
1. ไปที่ Configuration > Certificates > Controllers

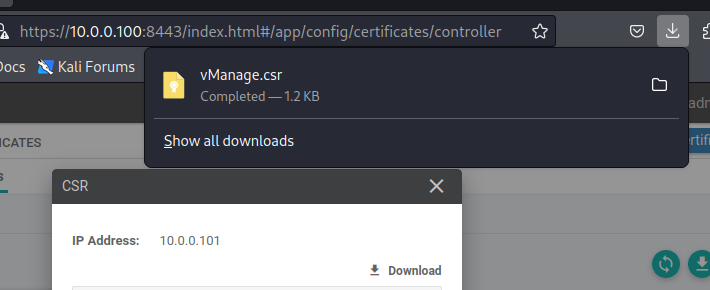


1. สำหรับแต่ละ Controller คลิก จุดสามจุด ทางขวา แล้วเลือก Generate CSR

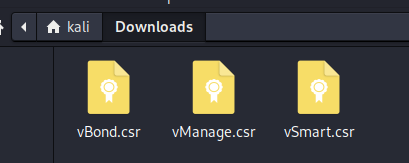


1. ดาวน์โหลดไฟล์ CSR ของแต่ละอุปกรณ์ และเปลี่ยนชื่อเป็น:
   1. vManage.csr

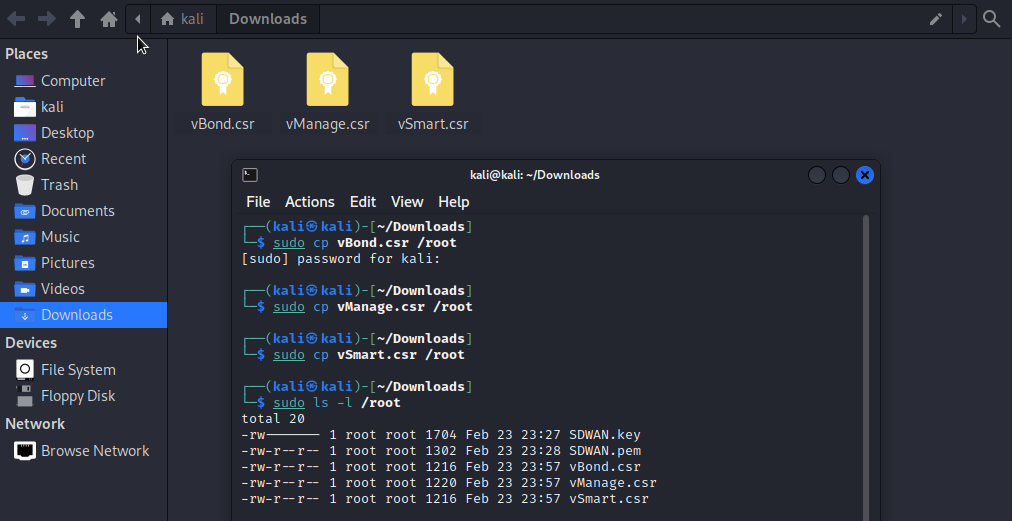




* 1. vBond.csr
  2. vSmart.csr



1. ย้ายไฟล์เหล่านี้ไปยังโฟลเดอร์เดียวกันกับ Root Key และ Root Certificate บน CA Machine



**สร้างและเซ็นใบรับรองอุปกรณ์ (Device Certificates)**

ใช้คำสั่งต่อไปนี้บน **CA Machine** เพื่อลงนาม (sign) ใบรับรองแต่ละตัว

openssl x509 -req -in vManage.csr -CA SDWAN.pem -CAkey SDWAN.key \

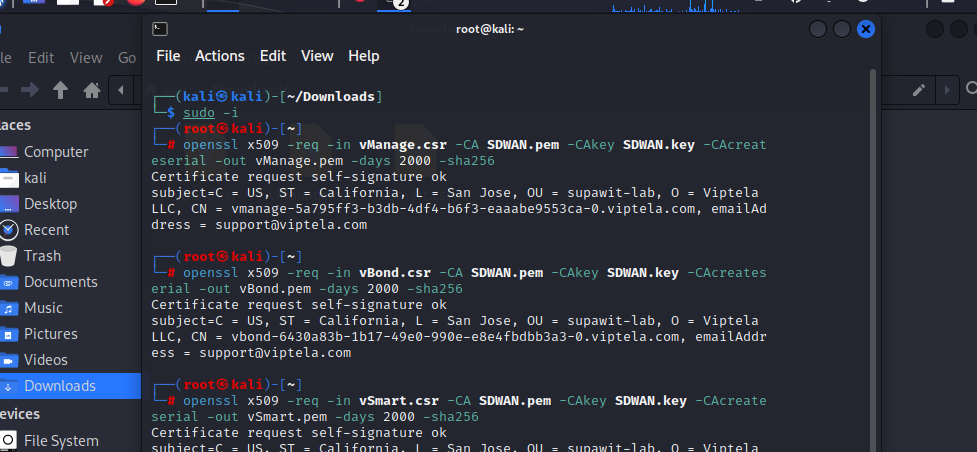
-CAcreateserial -out vManage.pem -days 2000 -sha256

openssl x509 -req -in vBond.csr -CA SDWAN.pem -CAkey SDWAN.key \

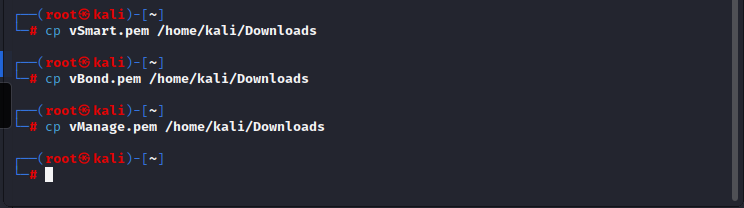
-CAcreateserial -out vBond.pem -days 2000 -sha256

openssl x509 -req -in vSmart.csr -CA SDWAN.pem -CAkey SDWAN.key \

-CAcreateserial -out vSmart.pem -days 2000 -sha256

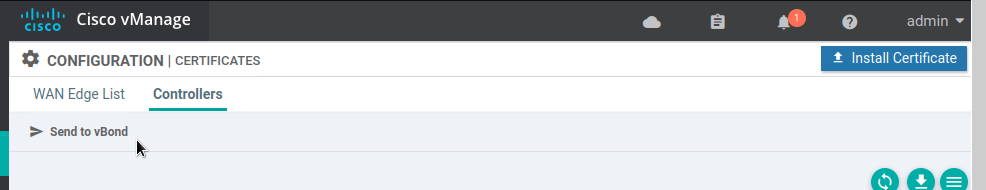


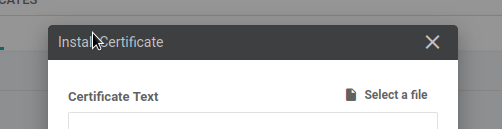
คัดลอกกลับไปยัง Downloads



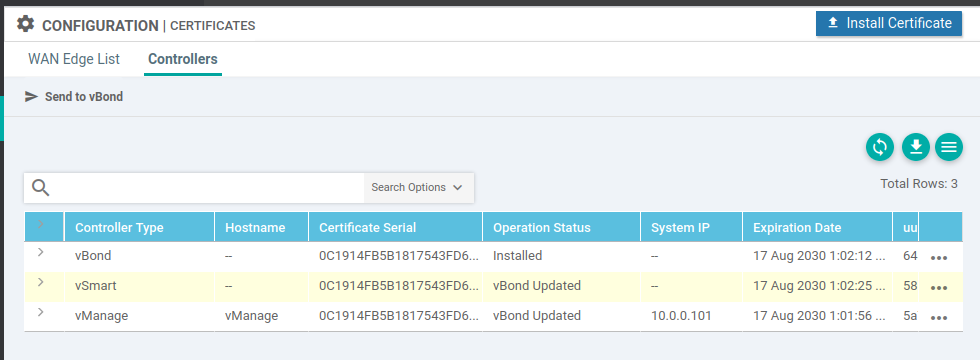
**ติดตั้งใบรับรองบน vManage**

1. ไปที่ Configuration > Certificates > Controllers
2. คลิก Install Certificate ที่มุมขวาบน





1. ติดตั้งไฟล์ Certificate ทั้งสามตัว
   1. vManage.pem
   2. vBond.pem
   3. vSmart.pem
2. หลังจากติดตั้งสำเร็จ กลับไปที่หน้า Controller Certificate Configuration
3. ตรวจสอบว่า Certificate Serial Number ของแต่ละอุปกรณ์ปรากฏขึ้นแล้ว



**เปิดใช้งาน VPN0 Tunnel Interface บน SD-WAN Controllers**

เปิดใช้งาน Tunnel บน vManage และ vSmart

ให้ล็อกอินเข้า Console ของ vManage และ vSmart แล้วป้อนคำสั่งนี้:

conf t

vpn 0

interface eth0

tunnel-interface

commit and-quit

ล็อกอินเข้า Console ของ vBond แล้วป้อนคำสั่งนี้:

conf t

vpn 0

interface ge0/0

tunnel-interface

encapsulation ipsec

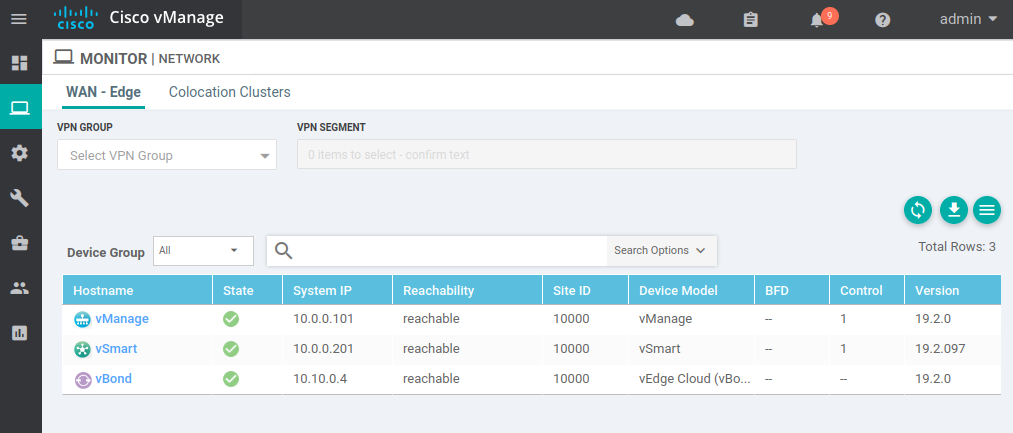
commit and-quit

**หลังจากตั้งค่าเสร็จ Dashboard จะอัปเดตเพื่อแสดงสถานะการเชื่อมต่อใหม่**

ตรวจสอบการเชื่อมต่อเพิ่มเติม

ไปที่ Monitor > Network และตรวจสอบว่า ทั้งสาม Controllers (vManage, vSmart, vBond) แสดงสถานะ "reachable"





#### cEdge01 (S100-CE1)

การนำอุปกรณ์ cEdge เข้าสู่ SD-WAN Fabric สำหรับ Site 100 ให้ล็อกอินเข้า Console ของ cEdge แล้วเพิ่มการตั้งค่าเริ่มต้น

config-transaction

!

hostname S100-CE1

username admin priv 15 secret admin

no ip domain lookup

!

system

system-ip 10.100.10.2

site-id 100

organization-name supawit-lab

vbond 10.10.0.3

exit

!

ip route 0.0.0.0 0.0.0.0 198.51.100.5

ip route 0.0.0.0 0.0.0.0 203.0.113.5

interface GigabitEthernet1

no shutdown

ip address 198.51.100.6 255.255.255.252

interface GigabitEthernet2

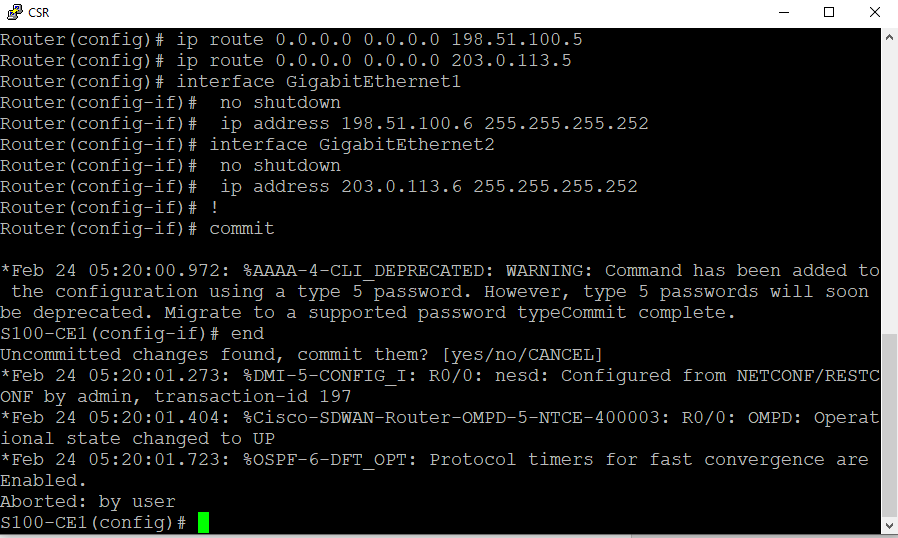
no shutdown

ip address 203.0.113.6 255.255.255.252

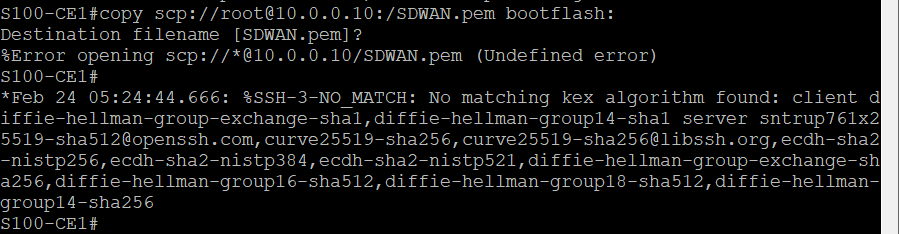
!

commit

end



เนื่องจาก Router นี้ ไม่ compatible ในเรื่อง key ssh กับตัว CA จึงต้องต้องใช้วิธีที่ซับซ้อนเล็กน้อย



**ไปที่ vManage cli**

1. เปิดไฟล์ SDWAN.pem ด้วย cat



1. จากนั้น copy เนื้อหาทั้งหมดใส่ใน format นี้ แล้ววางไว้ใน Notepad

tclsh

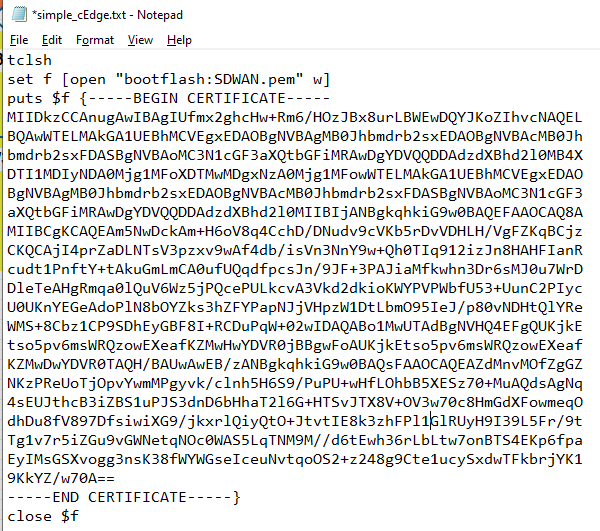
set f [open "bootflash:SDWAN.pem" w]

puts $f {-----BEGIN CERTIFICATE-----

(แทนที่ Key)

-----END CERTIFICATE-----}

close $f



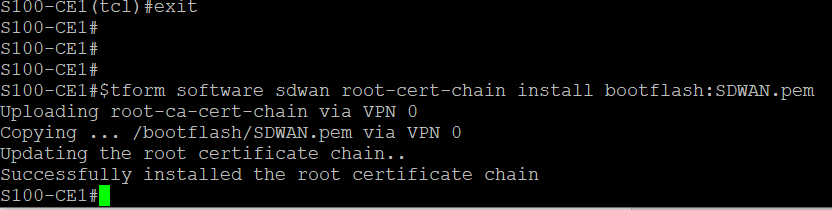
1. Copy คำเนื้อหาทั้งหมด ไปวางให้กับ cEdge

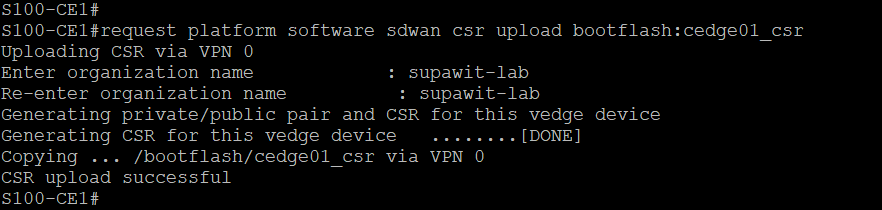


1. ใช้คำสั่งนี้กับ cEdge

request platform software sdwan root-cert-chain install bootflash:SDWAN.pem

request platform software sdwan csr upload bootflash:cedge01\_csr (supawit-lab, supawit-lab)



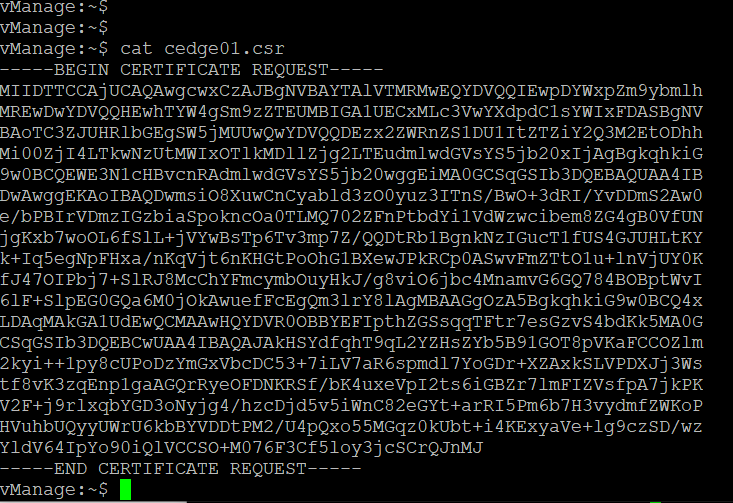


1. สร้าง cedge01.csr ที่ vManager ด้วย Vim จากนั้นวางเนื้อหาของ cedge01 ที่อยู่ใน cEdge01 ลงไป

vsh

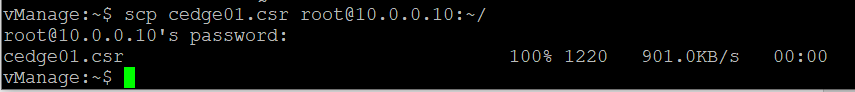
vi cedge01.csr





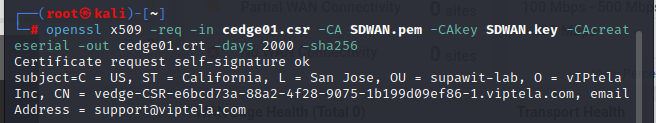
1. ส่ง cedge01.csr ไปให้ Linux (CA) แล้วทำการ Generate ด้วยคำสั่งนี้
   1. vManage cli

scp cedge01.csr root@10.0.0.10:~/



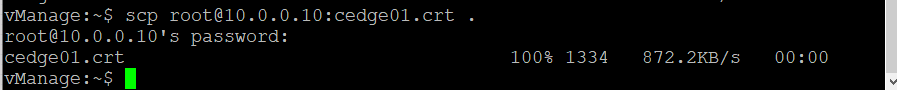
* 1. Linux (CA)

openssl x509 -req -in cedge01.csr -CA SDWAN.pem -CAkey SDWAN.key -CAcreateserial -out cedge01.crt -days 2000 -sha256

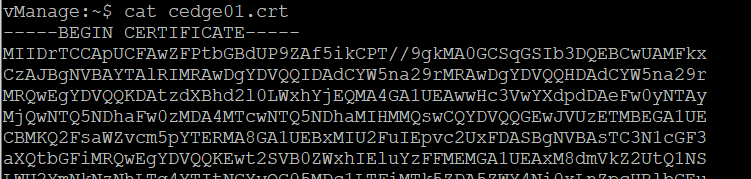


* 1. ใช้ vManage ดึง cedge01.crt กลับ

scp root@10.0.0.10:cedge01.crt .



* 1. ให้สร้างไฟล์ cedge01.crt ที่ cEdge01 โดยคัดลอกเนื้อหาใน cedge01.crt ที่ vManage ไปยัง cEdge01



tclsh

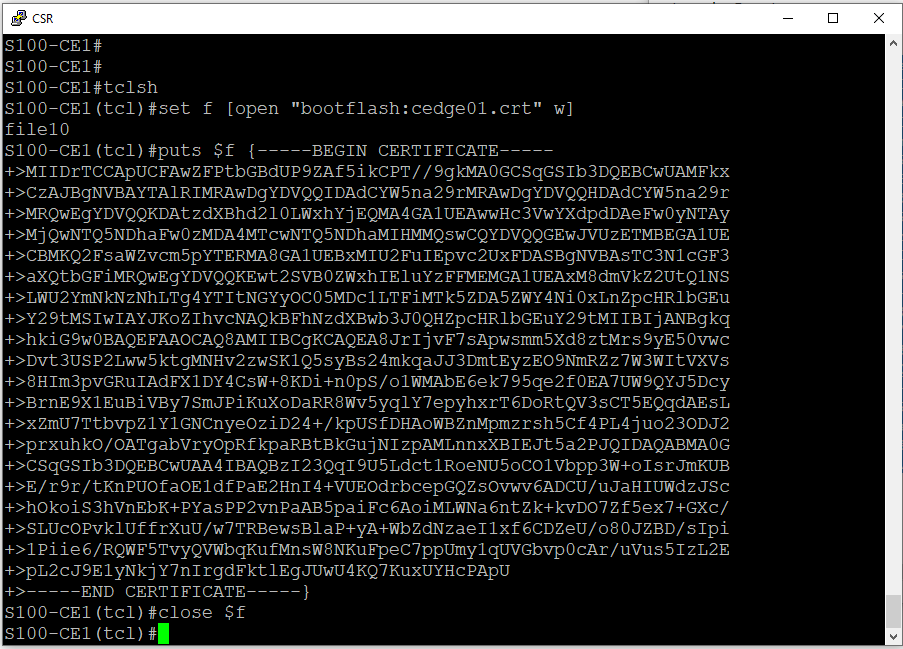
set f [open "bootflash: cedge01.crt" w]

puts $f {-----BEGIN CERTIFICATE-----

[แทนที่ key]

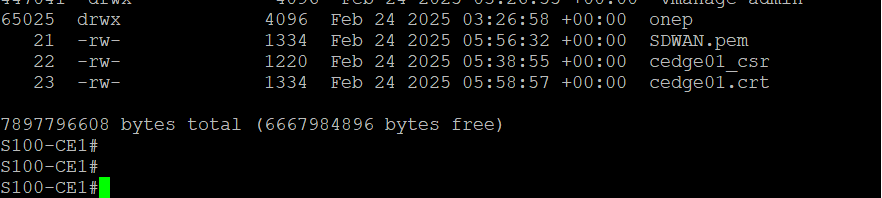
-----END CERTIFICATE-----}

close $f



ลองตรวจสอบด้วยคำสั่ง dir bootflash: จะต้องมีไฟล์ cedge01.crt

dir bootflash:

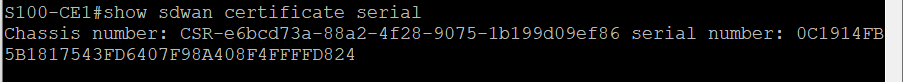


1. ลงทะเบียน Certificate ที่ cEdge01
2. ใช้คำสั่งนี้ลงทะเบียน

request platform software sdwan certificate install bootflash:cedge01.crt

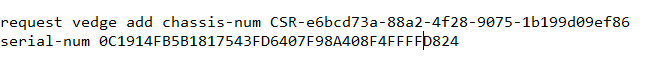
1. ตรวจสอบด้วย show sdwan certificate serial

show sdwan certificate serial

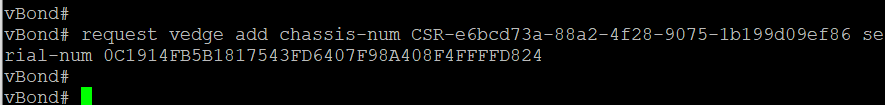


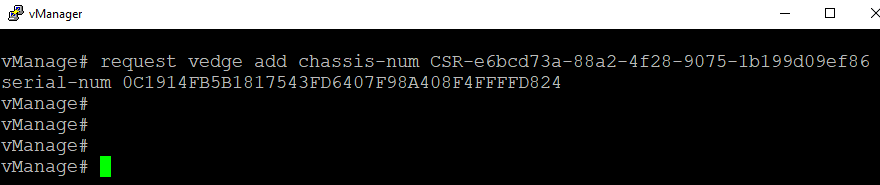
1. เปิด Notepad นำ chassis-num , serial number มาแทนที่คำสั่งนี้

request vedge add chassis-num [chassis-num] serial-num [serial-number]



1. นำคำสั่งที่ใส่ chassis-num และ serial ที่เตรียมไว้ ไปใช้กับ vManage & vBond





1. หลังจากติดตั้ง Root Certificate เสร็จสิ้นแล้ว กำหนดค่า Tunnel Interfaces : cEdge01

config-transaction

!

interface Tunnel 1

no shutdown

ip unnumbered GigabitEthernet1

tunnel source GigabitEthernet1

tunnel mode sdwan

exit

interface Tunnel 2

no shutdown

ip unnumbered GigabitEthernet2

tunnel source GigabitEthernet2

tunnel mode sdwan

exit

sdwan

interface GigabitEthernet1

tunnel-interface

encapsulation ipsec

color private1

exit

interface GigabitEthernet2

tunnel-interface

encapsulation ipsec

color public-internet

exit

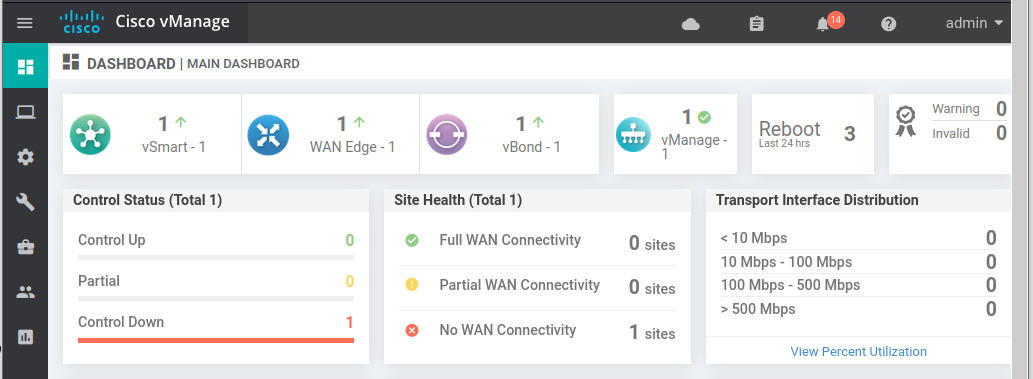
exit

!

commit

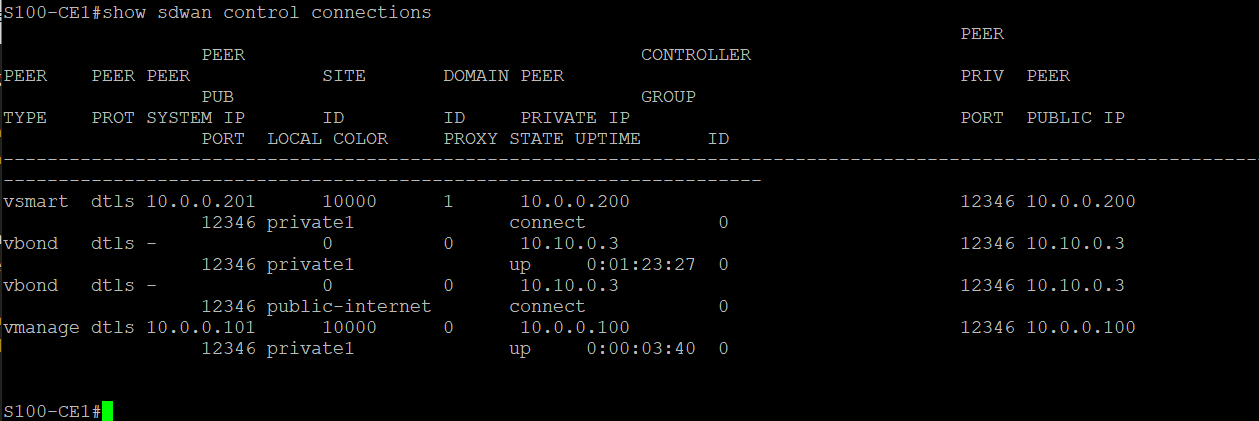
end

1. รอสักครู่แล้วตรวจสอบที่ vManage GUI



คำสั่งสำหรับตรวจสอบเบื้องต้น

show sdwan control connections



#### cEdge02 (S200-CE1)

ทำตามกระบวนการเดียวกันสำหรับ cEdge02

config-transaction

!

hostname S200-CE1

username admin priv 15 secret admin

no ip domain lookup

!

system

system-ip 10.200.10.2

site-id 200

organization-name supawit-lab

vbond 10.10.0.3

exit

!

ip route 0.0.0.0 0.0.0.0 198.51.100.9

ip route 0.0.0.0 0.0.0.0 203.0.113.9

interface GigabitEthernet1

no shutdown

ip address 198.51.100.10 255.255.255.252

interface GigabitEthernet2

no shutdown

ip address 203.0.113.10 255.255.255.252

!

commit

end

request platform software sdwan root-cert-chain install bootflash:SDWAN.pem

config-transaction

!

interface Tunnel 1

no shutdown

ip unnumbered GigabitEthernet1

tunnel source GigabitEthernet1

tunnel mode sdwan

exit

interface Tunnel 2

no shutdown

ip unnumbered GigabitEthernet2

tunnel source GigabitEthernet2

tunnel mode sdwan

exit

sdwan

interface GigabitEthernet1

tunnel-interface

encapsulation ipsec

color private1

exit

interface GigabitEthernet2

tunnel-interface

encapsulation ipsec

color public-internet

exit

exit

!

commit

end

**Summary :**

tclsh

set f [open "bootflash:SDWAN.pem" w]

puts $f {-----BEGIN CERTIFICATE-----

[key]

-----END CERTIFICATE-----}

close $f

request platform software sdwan root-cert-chain install bootflash:SDWAN.pem

request platform software sdwan csr upload bootflash:cedge02\_csr

copy to manager (vim cedge02.csr)more bootflash:/cedge02\_csr

send cedge02\_csr to kali

scp cedge02.csr root@10.0.0.10:~/

linux

openssl x509 -req -in cedge02.csr -CA SDWAN.pem -CAkey SDWAN.key -CAcreateserial -out cedge02.crt -days 2000 -sha256

ดึงกลับ

scp root@10.0.0.10:cedge02.crt .

cat cedge02.crt

tclsh

set f [open "bootflash:cedge02.crt" w]

puts $f {-----BEGIN CERTIFICATE-----

[key]

-----END CERTIFICATE-----}

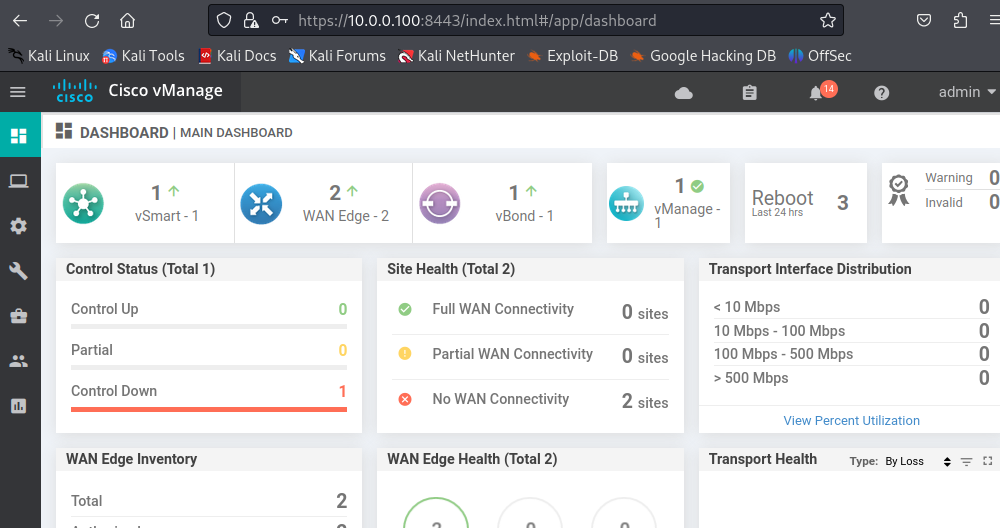
close $f

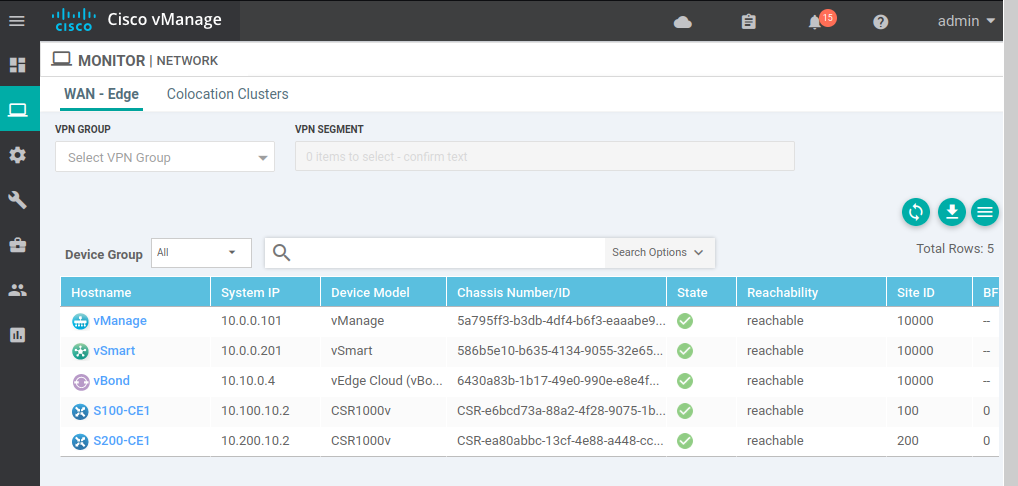
request platform software sdwan certificate install bootflash:cedge02.crt

show sdwan certificate serial

vbond & manage: request vedge add chassis-num CSR-ea80abbc-13cf-4e88-a448-cc4f112e77e1 serial-num 0C1914FB5B1817543FD6407F98A408F4FFFFD825

config tunnel





#### vEdge (DC1-VE1)

สุดท้ายนี้ เราจะนำ vEdge ของ Datacenter เข้าสู่ระบบ ซึ่งกระบวนการจะมีลักษณะคล้ายกัน

conf t

system

host-name DC1-VE1

system-ip 10.1.10.2

site-id 10000

organization-name supawit-lab

vbond 10.10.0.3

!

vpn 0

interface ge0/0

ip address 10.1.1.2/30

no tunnel-interface

no shutdown

ip route 0.0.0.0/0 10.1.1.1

!

commit and-quit

vshell

scp root@10.0.0.10:SDWAN.pem .

exit

request root-cert-chain install /home/admin/SDWAN.pem

conf t

vpn 0

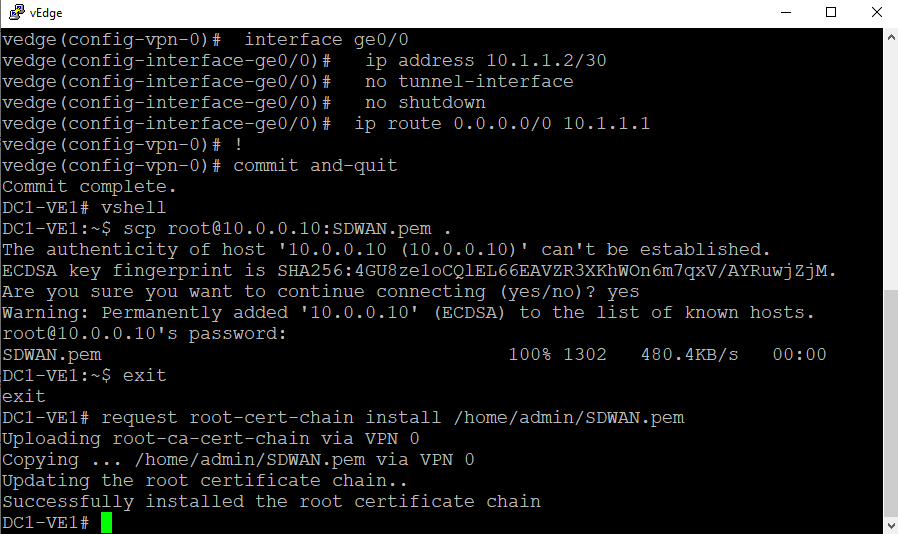
interface ge0/0

tunnel-interface

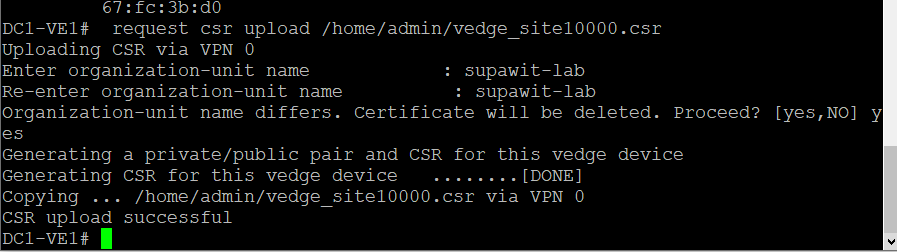
encapsulation ipsec

commit and-quit

1.



2. request csr upload /home/admin/vedge\_site10000.csr

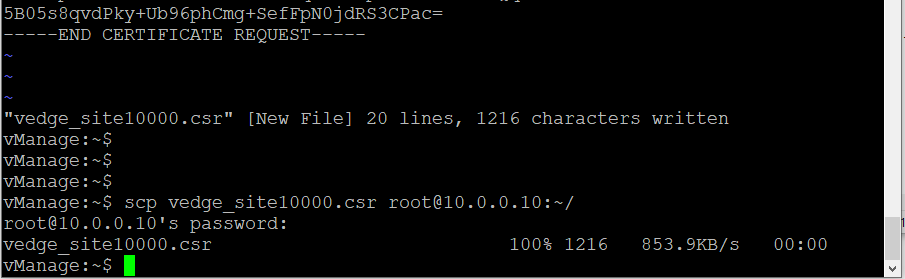


3. copy vedge\_site10000.csr from vEdge to vManage (vim)

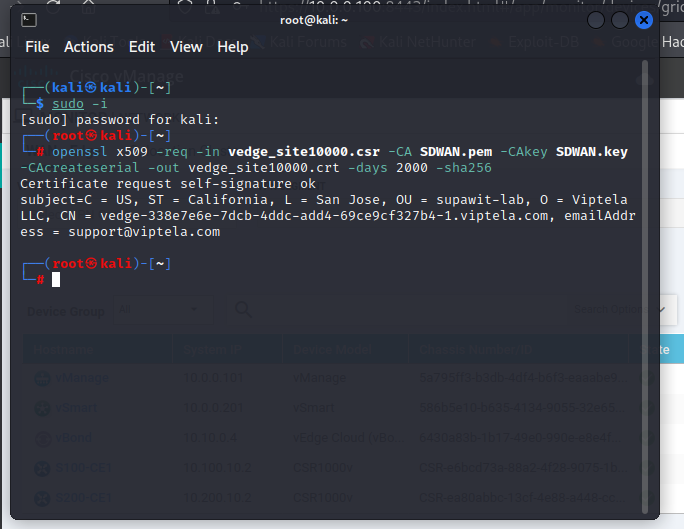


4. send vedge\_site10000.csr from vmanage to kali

scp vedge\_site10000.csr root@10.0.0.10:~/

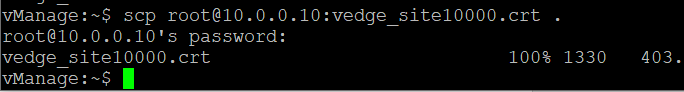


5. Create vedge\_site10000.crt

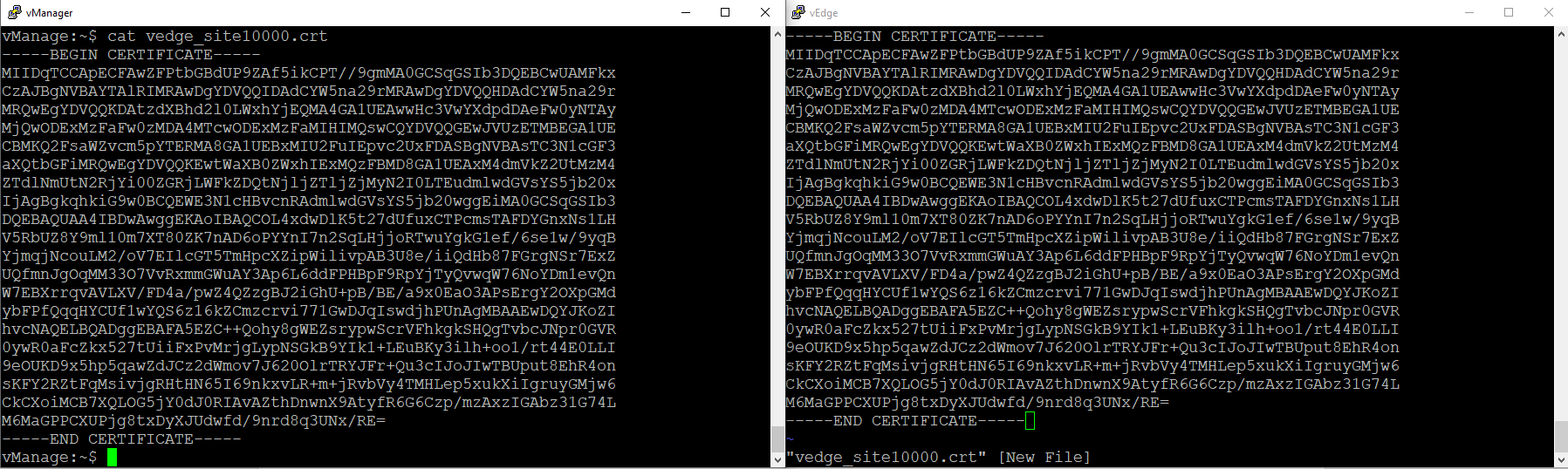


6. ดึงกลับ manage

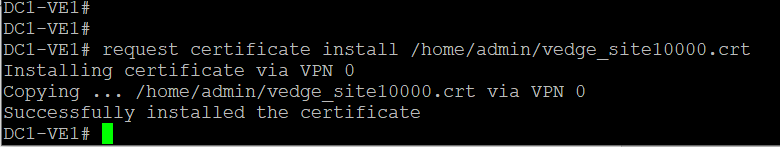
scp root@10.0.0.10:vedge\_site10000.crt .



7. copy vedge\_site10000.crt to vEdge



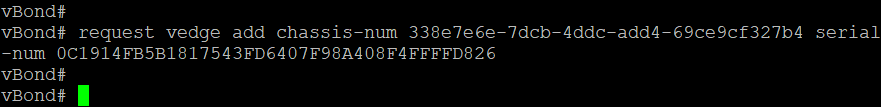
8. request certificate install /home/admin/vedge\_site10000.crt

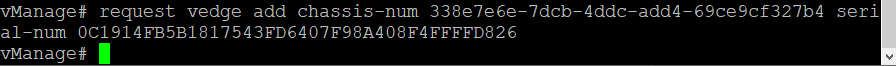


9. show certificate serial

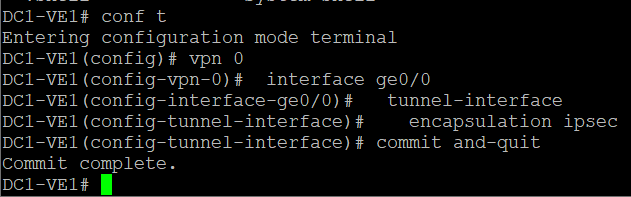
10. vmanage & vbond:

request vedge add chassis-num [-] serial-num [-]

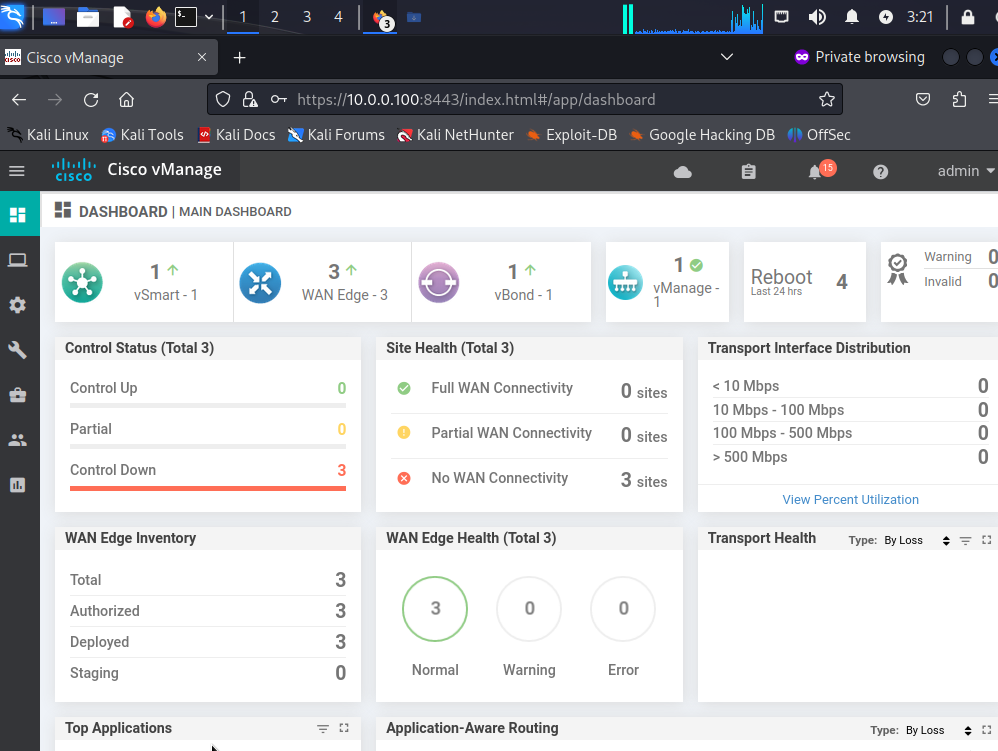


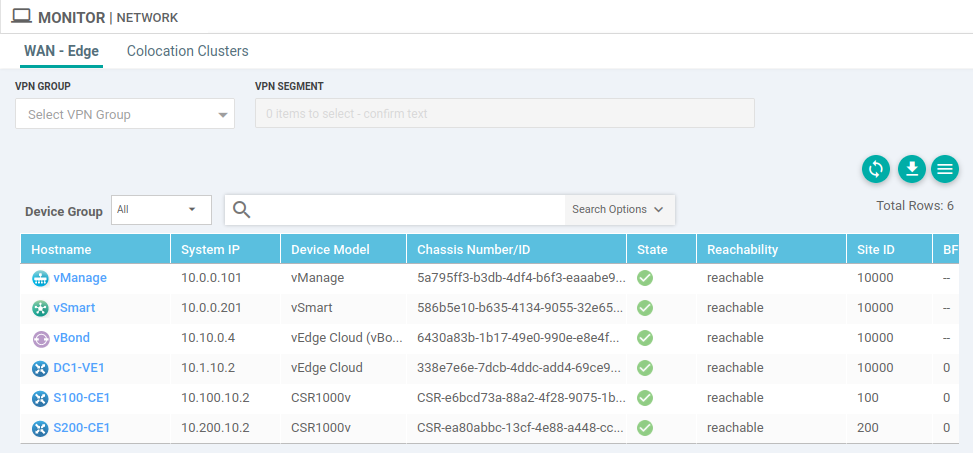


11. Config Tunnel



12. ตรวจสอบ Dashboard





13. Configuration > Certificates > Send to Controllers



#### DC1-SW2 & S100-SW1 & S200-SW1

host S100-SW1 #แก้ไขชื่อกับหมายเลข IP ให้ตรงตาม Topo

!

interface Ethernet0/0

switchport trunk encapsulation dot1q

switchport mode trunk

!

interface Ethernet0/1

!

interface Ethernet0/2

!

interface Ethernet0/3

!

vlan 10

vlan 20

vlan 30

!

interface Vlan10

ip address 10.100.10.254 255.255.255.0

no sh

!

interface Vlan20

ip address 10.100.20.254 255.255.255.0

no sh

!

interface Vlan30

ip address 10.100.30.254 255.255.255.0

no sh

!

ip route 0.0.0.0 0.0.0.0 10.100.10.1

ip route 0.0.0.0 0.0.0.0 10.100.20.1 5

ip route 0.0.0.0 0.0.0.0 10.100.30.1 10

!

end

#### Interface IPs:

* DC1-FW1:
  + g0/0 > 10.1.0.1 /30
  + g0/1 > 10.10.0.1 /24
  + g0/2 > 198.51.100.2 /30
  + g0/3 > 203.0.113.2 /30
* DC1-SW1:
  + VLAN 1 > 10.0.0.1 /24
  + VLAN 2 > 10.1.0.1 /30
  + VLAN 10 > 10.1.1.1 /30
* DC1-SW2:
  + VLAN 10 > 10.1.10.254 /24
  + VLAN 20 > 10.1.20.254 /24
  + VLAN 30 > 10.1.30.254 /24
* DC1-VE1:
  + ge0/0 > 10.1.1.2 /30
  + ge0/1.10 > 10.1.10.1 /24
  + ge0/1.20 > 10.1.20.1 /24
  + ge0/1.30 > 10.1.30.1 /24
* S100-CE1:
  + g1 > 198.51.100.6 /30
  + g2 > 203.0.113.6 /30
  + g3.10 > 10.100.10.1 /24
  + g3.20 > 10.100.20.1 /24
  + g3.30 > 10.100.30.1 /24
* S100-SW1:
  + VLAN 10 > 10.100.10.254 /24
  + VLAN 20 > 10.100.20.254 /24
  + VLAN 30 > 10.100.30.254 /24
* S200-CE1:
  + g1 > 198.51.100.10 /30
  + g2 > 203.0.113.10 /30
  + g3.10 > 10.200.10.1 /24
  + g3.20 > 10.200.20.1 /24
  + g3.30 > 10.200.30.1 /24
* S200-SW1:
  + VLAN 10 > 10.200.10.254 /24
  + VLAN 20 > 10.200.20.254 /24
  + VLAN 30 > 10.200.30.254 /24
* SP1:
  + g0/0 > 198.51.100.1 /30
  + g0/1 > 198.51.100.5 /30
  + g0/2 > 198.51.100.9 /30
  + g0/3 > 192.0.2.1 /30
* SP2:
  + g0/0 > 203.0.113.1 /30
  + g0/1 > 203.0.113.5 /30
  + g0/2 > 203.0.113.9 /30
  + g0/3 > 192.0.2.2 /30