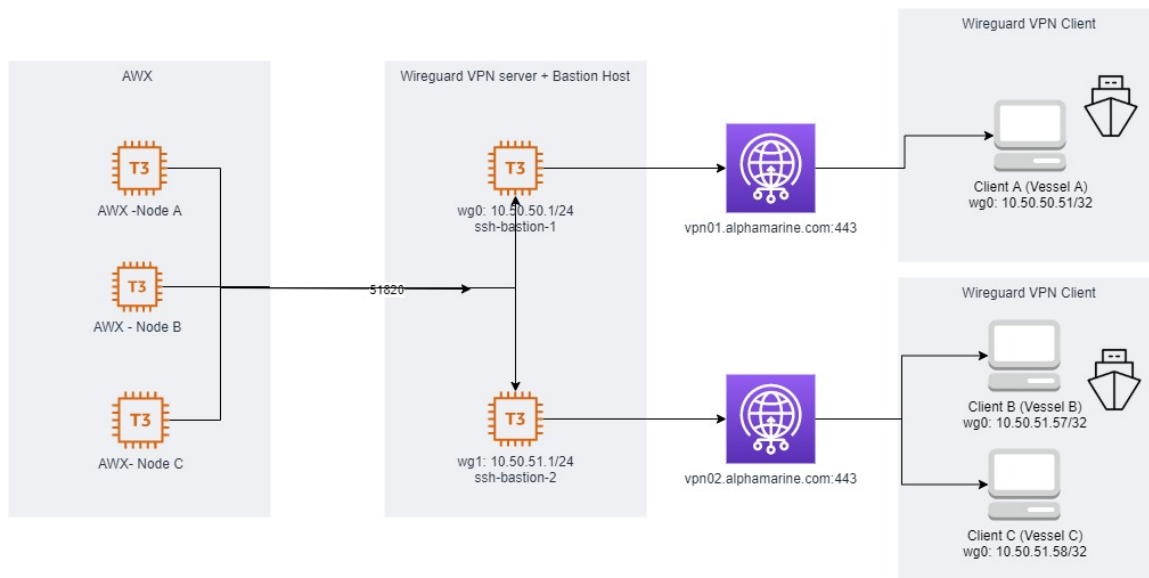


Wireguard Testing

Resources	Path
Dynamo DB vessel Ip details	https://us-east-2.console.aws.amazon.com/dynamodb/home?region=us-east-2#tables:selected=vessel_ip_details;tab=items
Ansible AWX	https://awx.alphaorimarine.com/#/jobs?job_search=page_size:20;order_by:-finished;not__launch_type:sync
wireguard test cases	https://docs.google.com/spreadsheets/d/1qwaVes6leSyR6ZI8pitszqn9XYpvlhE9xxrTxAtuqAc/edit?ts=5f55bd63#gid=0
Rundeck job for installation Token Generation	https://ops1.alphaorimarine.com/rundeck/project/AlphaOri/job/show/15ae7128-4399-47a3-885f-cbab118d2b3a
Wireguard architecture detailed	<p>The diagram illustrates the Wireguard architecture and installation process. It involves several AWS services and external tools:</p> <ul style="list-style-type: none"> Rundeck: Initiates the process by creating primary IPs in the <code>vessel_ip_pool</code> with a status of 0. It then creates a token for a specific ship primary node and updates the status to 1. Dynamo DB: Stores the IP details and token information. Lambda: Receives the token and public/private keys from the Ship Node Primary and verifies the vessel token. API Gateway: Acts as the interface for the Ship Node Primary to interact with the Lambda function. Ship Node Primary: Executes the installation script (<code>/home/centos/install.sh</code>) and provides the IMO number and token to the API Gateway. It also saves the configuration details in <code>/etc/wireguard/wg0.conf</code> and reboots to complete the installation. Primary Wireguard Server (Bastion Host): A central server with IP <code>vpn02.alphaorimarine.com:443</code> and <code>10.50.50.1</code> that maintains a secure connection with the Ship Node Primary. AWX: Used to execute commands in the ship server and receive responses. <p>The process flow is as follows:</p> <ol style="list-style-type: none"> Rundeck creates primary IPs in the <code>vessel_ip_pool</code> with used status 0. Rundeck creates a token for a specific ship primary node. Rundeck updates the status of the created IP to 1. The Ship Node Primary executes the installation script and provides the IMO number and token to the API Gateway. The API Gateway sends the token and public/private keys to the Lambda function. The Lambda function verifies the vessel token and sends the public/private keys back to the Ship Node Primary. The Ship Node Primary saves the configuration details in <code>/etc/wireguard/wg0.conf</code> and reboots to complete the installation. The Ship Node Primary maintains a secure connection with the Primary Wireguard Server (Bastion Host). AWX is used to execute commands in the ship server and receive responses.

Wireguard architecture for SMARTShip



Creating IP Pool and to verify in the Dynamo DB

The screenshot shows the AWS CloudFormation console for a stack named 'Demo Create Pool IP'. The stack is in the 'COMPLETE' state. The logs for the 'Script' step show the following output:

```

2014-11-24 10:50:50.10 already exists 54 06
2014-11-24 10:50:50.11 already exists 54 06
2014-11-24 10:50:50.12 already exists 54 06
2014-11-24 10:50:50.13 already exists 54 06
2014-11-24 10:50:50.14 already exists 54 06
2014-11-24 10:50:50.15 already exists 54 06
2014-11-24 10:50:50.16 successfully inserted in DB with type:primary
2014-11-24 10:50:50.17 successfully inserted in DB with type:primary
2014-11-24 10:50:50.18 successfully inserted in DB with type:primary
2014-11-24 10:50:50.19 successfully inserted in DB with type:primary
2014-11-24 10:50:50.20 successfully inserted in DB with type:primary
2014-11-24 10:50:50.21 successfully inserted in DB with type:primary
2014-11-24 10:50:50.22 successfully inserted in DB with type:primary
2014-11-24 10:50:50.23 successfully inserted in DB with type:primary
  
```

Below the logs, the 'Outputs' tab shows the 'ip_address' output, which is a list of IP addresses. The table below shows the details of the IP addresses created:

ip_address	type	status	count
10.50.50.10	OK	successfully	1
10.50.50.11	OK	successfully	1
10.50.50.12	OK	successfully	1
10.50.50.13	OK	successfully	1
10.50.50.14	OK	successfully	1
10.50.50.15	OK	successfully	1
10.50.50.16	OK	successfully	1
10.50.50.17	OK	successfully	1
10.50.50.18	OK	successfully	1
10.50.50.19	OK	successfully	1
10.50.50.20	OK	successfully	1
10.50.50.21	OK	successfully	1
10.50.50.22	OK	successfully	1
10.50.50.23	OK	successfully	1

As per the IP address count, 10.50.50.0/26, 64 address location should be created. But actually, it was only 54(6 already created + 48 created new + 10 reserved IP addresses)

Token Generation, if the vessel already exists

ops1.alphaorimarine.com/rundeck/project/AlphaOri/execution/show/719961#nodes

AlphaOri

Vessel On-boarding

✓ Demo SMARTShip Installation Token Generator

Options:
Vessel: IRON MIRACLE-9500742
nodetype: primary

[Log Output »](#)

100% 1/1 COMPLETE 0 FAILED 0 INCOMPLETE 0 NOT STARTED

Node	Start time	Duration
localhost		0.00:02
Script		0.00:02
00:14:06 IRON MIRACLE		
00:14:06 10.50.50.12		
00:14:06 9500742		
00:14:06 Node IP (10.50.50.12) already assigned for vessel no 9500742For wireguard primary server. If you want to reinstall use below token.		
00:14:06 Node IP: 10.50.50.12		
00:14:06 NEW Token for SMARTShip Installation is generated:3968E123		
00:14:06 Token generated at Sep-07-2020 18:44:05UTC time and valid for 15 minutes.		

New Token Generated for assinged IP for 9500742



Rundeck Token Generator <no-reply@sns.amazonaws.com>
To sajan@alphaori.sg

[↩ Reply](#)

[↩ Reply All](#)

[→ Forward](#)



Tue 08-09-2020 00:14

NEW Token for SMARTShip Installation is generated. For Node IP:10.50.50.12

--

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:

<https://sns.us-east-2.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-2:945995963191:awxnotifications:4c390902-d7fb-4fd1-8de9-680103bdaa8d&Endpoint=sajan@alphaori.sg>

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at <https://aws.amazon.com/support>

Token
Generation,
for new
vessel

ops1.alphaorimarine.com/rundeck/project/AlphaOri/execution/show/719964#nodes

Vessel On-boarding

✓ Demo SMARTShip Installation Token Generator

Options:
Vessel:
FURANO GALAXY-9804928
nodetype:
primary

Succeeded 0.00:1
you

Log Output »

100% 1/1 COMPLETE 0 FAILED 0 INCOMPL

Node

localhost All Steps OK

Script OK

00:16:33 FURANO GALAXY
00:16:33 10. 50. 50.14
00:16:33
00:16:33 9804928
00:16:33 Token for SMARTShip Installation is 4A555426
00:16:33 Token generated at Sep-07-2020 18:46:32UTC time and valid for 15 minutes.
00:16:33 Updated client 10.50.50.14 IP on wireguard Primary server

Scan: [Table] vessel_ip_details: ref_key ^ Viewing 1 to 5 items

Scan [Table] vessel_ip_details: ref_key ^
+ Add filter
Start search

ref_key	details	timestampcheck	vessel_no
1599475617	{ "nodeip" : { "S" : "10.50.50.10" }, "nodename" : { "S" : "BW HAZEL" }, "nod...	1599475617	9626687
1599476484	{ "nodeip" : { "S" : "10.50.50.11" }, "nodename" : { "S" : "BW CANOLA" }, "n...	1599479220	9687124
1599479982	{ "nodeip" : { "S" : "10.50.50.12" }, "nodename" : { "S" : "IRON MIRACLE" }, ...	1599504245	9500742
1599480800	{ "nodeip" : { "S" : "10.50.50.13" }, "nodename" : { "S" : "BW YUSHI" }, "nod...	1599480800	9810044
1599504392	{ "nodeip" : { "S" : "10.50.50.14" }, "nodename" : { "S" : "FURANO GALAXY..."	1599504392	9804928

New Token generated for FURANO GALAXY



Rundeck Token Generator <no-reply@sns.amazonaw
To sajan@alphaori.sg

Reply

Reply All

Forward



Tue 08-09-2020 00

Check Rundeck job for new Token generated at Sep-07-2020 18:46:32UTC time and valid for 15 minutes.

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:

<https://sns.us-east-2.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-2:945995963191:awxnotifications:4c390902-d7fb-4fd1-8de9-680103bdaa8d&Endpoint=sajan@alphaori.sg>

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at <https://aws.amazon.com/support>

Useful
commands

```
systemctl restart wg-quick@tun6.service
sudo systemctl status wg-quick@wg0      # to check whether the
wireguard is running
vi /etc/wireguard/wg0.conf                # path for the
wireguard config file
sh /home/centos/install.sh                #to setup the
wireguard configuration for a particular vessel
```

systemctl restart wg-quick@tun6.service

install.sh
script

```
read -p "Enter Vessel IMO Number: " IMO
read -p "Enter Installation Token: " TOKEN
#MAC = `cat /sys/class/net/eth0/address`
KEY=$(curl -sb -H "Accept: application/json"
"https://ldlpq5w40k.execute-api.ap-south-1.amazonaws.com
/VerifyVesselToken?vessel=$IMO&token=$TOKEN&mac=$MAC" | grep -
i "ary")

if [[ "$KEY" == "false" ]]; then
    echo "Invalid IMO Number or Token! Installation
Failed!!"
    exit
fi

IFS='|' read -r -a ADDR <<< "$KEY"
NODE_IP=${ADDR[0]}
NODE_NAME=${ADDR[1]}
NODE_TYPE=${ADDR[2]}
CLIENT_KEY=${ADDR[3]}
CLIENT_PUB_KEY=${ADDR[4]}
WG_PUB_KEY=${ADDR[5]}

sudo yum install epel-release https://www.elrepo.org
/elrepo-release-7.el7.elrepo.noarch.rpm -y
#sudo yum install yum-plugin-elrepo
sudo yum install wireguard-dkms wireguard-tools kmod-
wireguard -y

#sudo sh -c 'wg genkey | tee privatekey | wg pubkey >
publickey'
sudo mkdir /etc/wireguard
if [[ "$NODE_TYPE" == "primary" ]];
then
sudo -- bash -c '/bin/cat <<EOF >/etc/wireguard/tun6.conf
[Interface]
Address = wg_client_ip/32
```

```

PrivateKey = client_private_key

[Peer]
PublicKey = server_public_key
AllowedIPs = 10.50.40.1/24
Endpoint = stagevpn01.alphaorimarine.com:443
PersistentKeepalive = 15

EOF'
fi
if [[ "$NODE_TYPE" == "secondary" ]];
then
sudo -- bash -c '/bin/cat <<EOF >/etc/wireguard/tun6.conf
[Interface]
Address = wg_client_ip/32
PrivateKey = client_private_key

[Peer]
PublicKey = server_public_key
AllowedIPs = 10.50.41.1/24
Endpoint = stagevpn02.alphaorimarine.com:443
PersistentKeepalive = 15


EOF'
fi

sudo sed -i "s/wg_client_ip/$NODE_IP/" /etc/wireguard/tun6.
conf
sudo sed -i "s%client_private_key%$CLIENT_KEY%" /etc
/wireguard/tun6.conf
sudo sed -i "s%server_public_key%$WG_PUB_KEY%" /etc
/wireguard/tun6.conf

sudo systemctl enable wg-quick@tun6.service

read -p "Reboot machine now? " REBOOT_CONFIRM
if [[ "$REBOOT_CONFIRM" != "Y" ]]; then
echo "Reboot is required! You may please reboot
manually"
exit
fi
sudo reboot

```

stage wireguard install.sh script	
--	---

Footnotes

Term	Meaning/Resource
Global Accelerator	<p>Improve global application availability</p> <p>AWS Global Accelerator continually monitors the health of your application endpoints, such as your Network Load Balancers, Application Load Balancers, EC2 Instances, or Elastic IPs, instantly reacting to changes in their health or configuration.</p>
WireGuard	<p>WireGuard is a free and open-source software application and communication protocol that implements virtual private network techniques to create secure point-to-point connections in routed or bridged configurations</p>
Ansible Playbook	<p><i>Playbooks are Ansible's configuration, deployment, and orchestration language. They can describe a policy you want your remote systems to enforce, or a set of steps in a general IT process.</i></p> <p><i>If Ansible modules are the tools in your workshop, playbooks are your instruction manuals, and your inventory of hosts are your raw material.</i></p> <p>Playbooks are the files where Ansible code is written. ... Playbooks are one of the core features of Ansible and tell Ansible what to execute. They are like a to-do list for Ansible that contains a list of tasks. Playbooks contain the steps which the user wants to execute on a particular machine.</p>
bastion host	<p>A bastion host is a special-purpose computer on a network specifically designed and configured to withstand attacks. The computer generally hosts a single application, for example a proxy server, and all other services are removed or limited to reduce the threat to the computer.</p>
CIDR	<p>https://cidr.xyz/</p> <p>Classless_Inter-Domain_Routing</p> <p>Classless inter-domain routing (CIDR) is a set of Internet protocol (IP) standards that is used to create unique identifiers for networks and individual devices. The IP addresses allow particular information packets to be sent to specific computers.</p>
Why do we need to create a pool of IPs in he beginning itself. why can't we create it in runtime when in need?	<ul style="list-style-type: none"> On each peer, create a WireGuard interface and assign an IP address to it with the <code>ip</code> tool. It's an address inside a VPN network bound to the peer forever On each peer generate a private key using the <code>wg</code> tool and assign it to the WireGuard interface Derive a public key, again with the <code>wg</code> tool, and add it to all other peers you want to communicate. WireGuard doesn't specify how to exchange the keys. I opened an SSH session on each device and copied them over manually Optionally, tell each peer how to reach other peers by specifying a public IP (or domain) and a port. Not all peers need to know how to reach others, as long as others know how to reach them; you'll see later

References

Resource	Path
Wireguard real world usage scenarios	https://www.zahradnik.io/wireguard-a-vpn-with-real-world-usage-in-mind
Wireguard use cases and commands	https://www.ivpn.net/knowledgebase/255/Linux---Autostart-WireGuard-in-systemd.html
wireguard comparison	https://www.ivpn.net/pptp-vs-ipsec-ikev2-vs-openvpn-vs-wireguard