appgate

Appgate SDP for GCP

— REFERENCE ARCHITECTURES

Type: Technical guide

Date: April 2020

Applies to: Appgate SDP v4.3 or newer

TABLE OF CONTENTS

INTRODUCTION

As businesses accelerate their adoption of GCP for high-value production workloads, the current threat and compliance landscape demands that security be considered up-front, and in fact act as an enabler, rather than an impediment to user productivity and business agility. The Software-Defined Perimeter (SDP)—a new, open security architecture—is ideally-suited to securing exactly the kind of distributed, dynamic, and API-driven environments in which GCP excels, and enables agility while improving user productivity.

Appgate SDP is Appgate product built on the Software-Defined Perimeter specification; implementing the core SDP principles as outlined by the Cloud Security Alliance's SDP Working Group, coupled with a few valuable extensions to support enterprise scale and operational management requirements. The most unique aspect of SDP is that while access policies are defined for identities, enforcement functions at the network layer, this provides direct connectivity from the user to multiple protected network resources while responding to environmental conditions and user attributes in real-time.

Appgate SDP has some very specific advantages over traditional legacy infrastructures:

User-Centric Network Security: Appgate SDP provides application and service-specific authentication and authorization to uniquely grant network access from within and outside of the corporate perimeter. Appgate SDP dynamically creates a secure, encrypted network "segment of one" that is tailored on user's specific attributes for each user session. Unlike traditional approaches, network access rules are not static and unchanged for months or years, but are dynamically instantiated and enforced.

Cloud-Native: Appgate SDP is designed to support IaaS environments—with a flexible, distributed deployment model which suits many different cloud architectures; Appgate SDP automatically detects server instance creation, and leverages user and server metadata to evaluate access. Driven by a common policy model, Appgate SDP orchestrates these elements—dynamically controlling access by authenticated users to specified cloud resources.

Seamless Integrations: Appgate SDP reduces costs by eliminating IP address configuration, ad-hoc third-party set-up, and managing user access across a hybrid cloud infrastructure security. Appgate combines authorization, encryption and access control in one system while seamlessly integrating with existing identity, multi factor authentication and SIEM solutions. Its API-first architecture enables businesses to utilize existing authentication, logging, and incident response processes to quickly support agile hybrid cloud security requirements into their operations and security processes.

Security On-Demand: Appgate SDP is built on a distributed model to support a variety of use cases and provide an architecture which aligns with the security controls of the hosts and applications. This ensures traffic is encrypted over any network used to access the resources are in close proximity, eliminating the potential security risk presented by the intermediate network.

Compliance is Key: Appgate SDP helps enterprises reduce regulatory compliance costs by reducing scope and audit complexity. Cloud providers offer some functionality for the myriad regulatory requirements, but Appgate SDP can greatly enhance these by providing a common logging and federated access framework. With this in place, Appgate SDP inherently reduces the number systems that fall within audit scope through its approach—often eliminating the need for regulatory controls themselves. Robust 360-degree, user-centric logging also provides any evidence necessary to meet audit requirements.

Hybrid Cloud: Even as organizations migrate to GCP, they often have on-premises resources. Appgate SDP's architecture and policy model supports access control for cloud and on-premises resources from a single, integrated platform.

The remainder of this document introduces the Software-Defined Perimeter architecture and an overview of several GCP reference architectures for Appgate SDP. For further information about Appgate SDP, please visit: https://www.appgate.com/software-defined-perimeter.

WHAT IS THE SOFTWARE-DEFINED PERIMETER?

The Software-Defined Perimeter is a very different approach to the problem of securing today's networks. Its aim is to solve the problem of stopping network attacks on application infrastructure, while ensuring user productivity and improving security operations efficiency. The CSA SDP Working Group developed a clean-sheet approach that combined on-device authentication, identity-based access and dynamically provisioned connectivity.

Specifically, Appgate SDP is comprised of five distinct functions to not only support the tenets of the SDP specification, but to enable high availability and fit within the operational framework of an enterprise.

Controller: Acts as the brains of the Appgate SDP platform and is the acting control plane management function for orchestrating Policies. These decisions may be granted simply based on IDP authentication or based on more complex decisions involving Conditions, device posture, or 3rd party information.

Gateway: Connects the user's device session to the resource. The Gateway is designed for carrier-grade high availability and throughput to meet the most demanding use cases.

Client: Installed on user devices, this component securely establishes an encrypted, tunnelled network connection to the Appgate SDP Gateways, ensuring that all user traffic is secured. This component also performs device inspection and posture checks, to enforce access policies.

IoT Connector: Provides ability to extend SDP to business premises, whether branch office or other remote location in order to securely enforce policies based on device classification or observed behaviours.

LogServer/LogForwarder: Appgate SDP generates detailed and searchable user-centric access logs, which are valuable for security and compliance purposes. These logs can be stored within the Appgate SDP system or forwarded to an enterprise SIEM.

While many of the security components in SDP are well-proven, the integration of these components is quite novel. More importantly, the SDP security model has been shown to stop network attacks including DoS, Man-in-the-Middle, Server Query (OWASP10) as well as Advanced Persistent Threats (APT). SDP was not designed as another DMZ add-on to an existing set of security controls such Proxies or VPNs. Rather, this new model provides an alternative to these outdated tools which were developed in a time before (hybrid) Cloud became all pervasive.

It is important to understand that Appgate SDP has filled in the gaps in the CSA's SDP model and extended it with specific enhancements, built on several key principles:

The first principle is that users and devices operate outside the traditional perimeter. Today's user populations are diverse, and operate from many different physical locations. And, they are accessing server resources that are more and more located in the Cloud (or at least remotely from the users)

The second principle of a Software-Defined Perimeter is the notion of "authenticate first, connect second". Unlike a traditional network that connects users in various roles or groups to a network segment and then relies on application level permissions for authorization, a Software-Defined Perimeter creates individualized permissions; as a user's situation changes, the individualized permissions changes. This ensures that only authorized users can connect to network resources. Resources are rendered invisible to dangerous reconnaissance which greatly reduces the attack surface and significantly enhances a corporation's security posture.

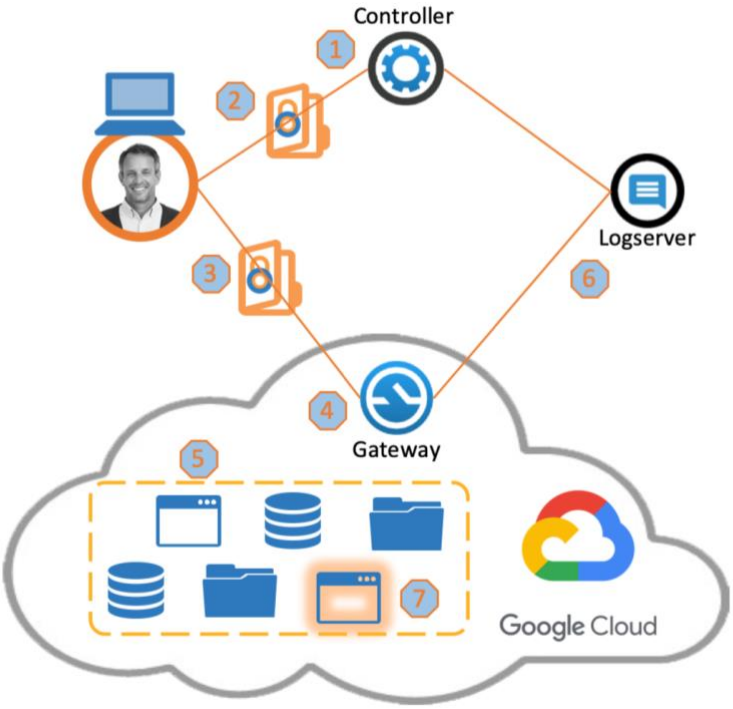
The third principle is that the access controls should be placed as close to the protected hosts as possible. When the user attempts to access a resource—for example by opening a web page on a protected server, the Client redirects the request to the closest Gateway via a secure tunnel. This in turn applies additional policies in real time—for example, to control access based on the user's network location. This approach allows Clients to make multiple connections to multiple Gateways simultaneously across different clouds if necessary, to address the user's specific connectivity needs.

Having multiple Gateways (access points) makes the SDP very suitable for hybrid environments—allowing consistent access policies to be applied to legacy network, data center and cloud environments. New Sites are very independent of one another and easily deployed with no long lead-times; they simply require Internet access.

CORE APPGATE SDP ARCHITECTURE

Appgate SDP draws on user context to dynamically create a secure, encrypted network 'segment of one' that's tailored for each user session.

It works as follows:



1. User authenticates to the Controller
2. Controller applies Policies based on user claims, roles, and context, and issues a signed token listing the resources the user is entitled to
3. User attempts to access a protected resource behind a Gateway
4. Gateway evaluates any Entitlements in real-time, ensuring that all Conditions are met—for example network location, time of day, device health, and service metadata, such as security groups. Users may be prompted for additional information, such as a one-time password
5. If permitted, the Gateway opens a connection to the target resource for the user
6. LogServer provides secure logging services
7. Gateway automatically detects any changes to the Cloud services using metadata and adjusts user access based on the Entitlements.

Appgate SDP can be deployed into GCP in a number of different ways. Before exploring the different deployment scenarios, it is worth just covering a few points which apply to all the different scenarios that follow:

Connectivity

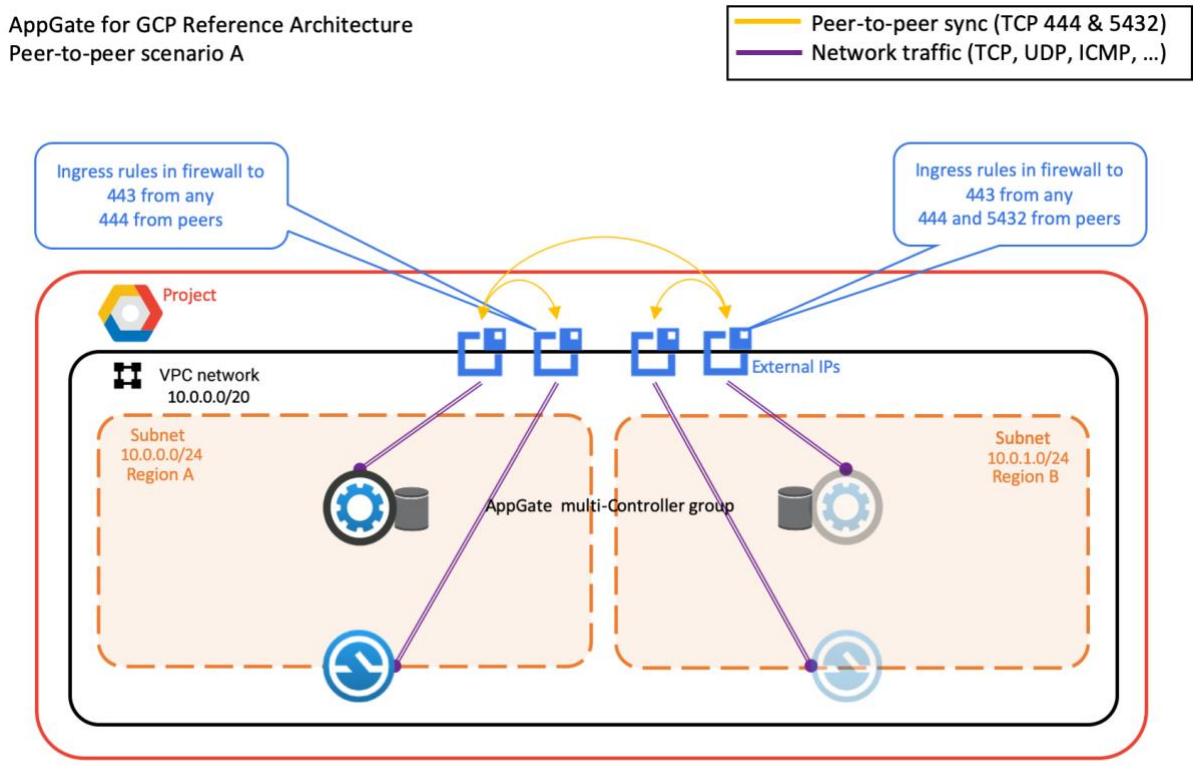
Clients will typically use multiple DNS A records (round-robin) to connect to a multi-Controller group using TLS. It would also possible to use GCP Network services — Load balancing as an alternative to distribute the traffic to one of the clustered Controllers but this is not explored in these scenarios.

Clients self-manage the connections to an available Gateways using mutual TLS on port 443.

Peer-to-peer communication between Controllers use mutual TLS traffic on port 444 and 5432. This will typically happen via the external IPs and is not shown in the scenarios.

Peer-to peer communication between Controllers and Gateways uses mutual TLS traffic on port 444. This will typically happen via the external IPs and is not shown in the scenarios. There is no communication between Gateways whatsoever.

It is recommended that peer-to-peer communication is limited to ONLY the IP addresses of the peers.



Using the GCP network

Where the Appgate SDP Collective is deployed inside GCP then it might be a good idea to route all the peer-to-peer traffic internally rather than via the external IPs (over the internet). This can be very easy to manage as the GCP concept of the VPC is global — so it really does not matter where an appliance is deployed. There can be several advantages to this:

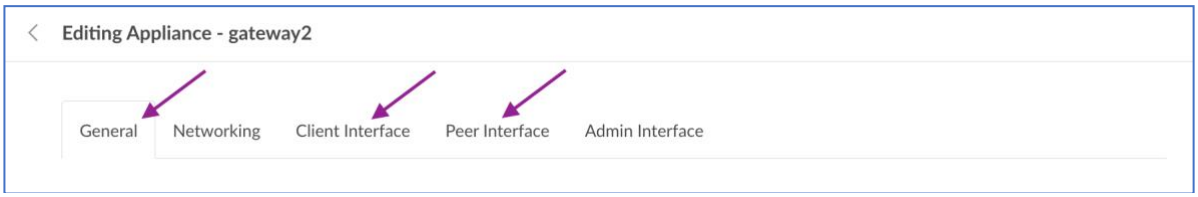
Security: peer traffic and ports are no longer exposed to the internet

Performance: peer traffic is all routed over the GCP internal network

Cost: there are no ingress/egress charges

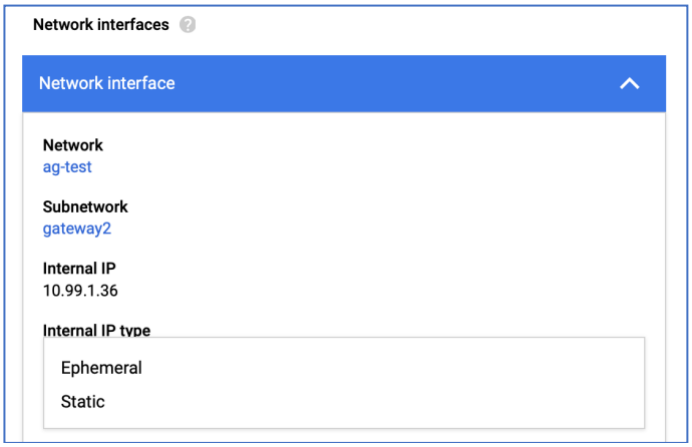
DoS: it is no longer possible to perform DoS on the peer ports

The TLS peer-to-peer communication is based on appliance certificates which pick up hostname(s) of the appliances (or IP addresses) from the appliance configuration. These hostnames are set in 3 places System>Appliances>General, System>Appliances>Client Interface, System>Appliances>Peer Interface.

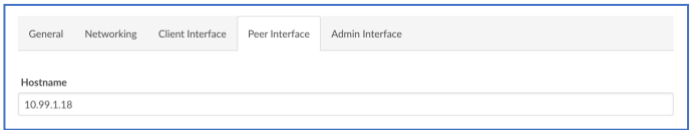


Using internal IP addresses

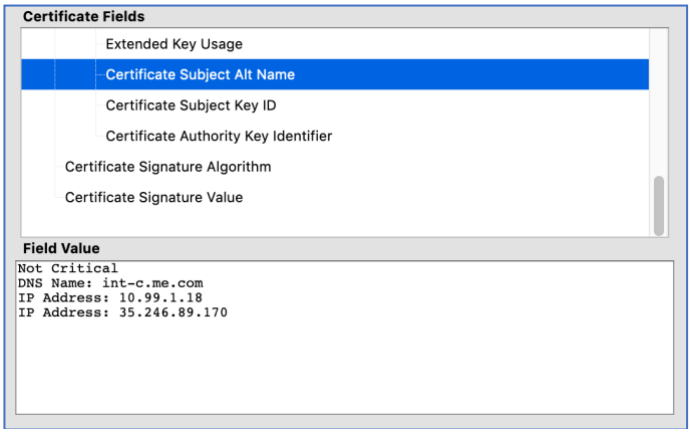
GCP has a very nice facility which allows you to reserve the internal IP address which the appliance is given by DHCP. So, after the appliance is started you can change the Ephemeral IP address to a Static one (or you can do this beforehand).



Appgate SDP appliances will talk to each other using the hostname in the Peer Interface tab. Once you know your static IP then set this as the Peer interface hostname.

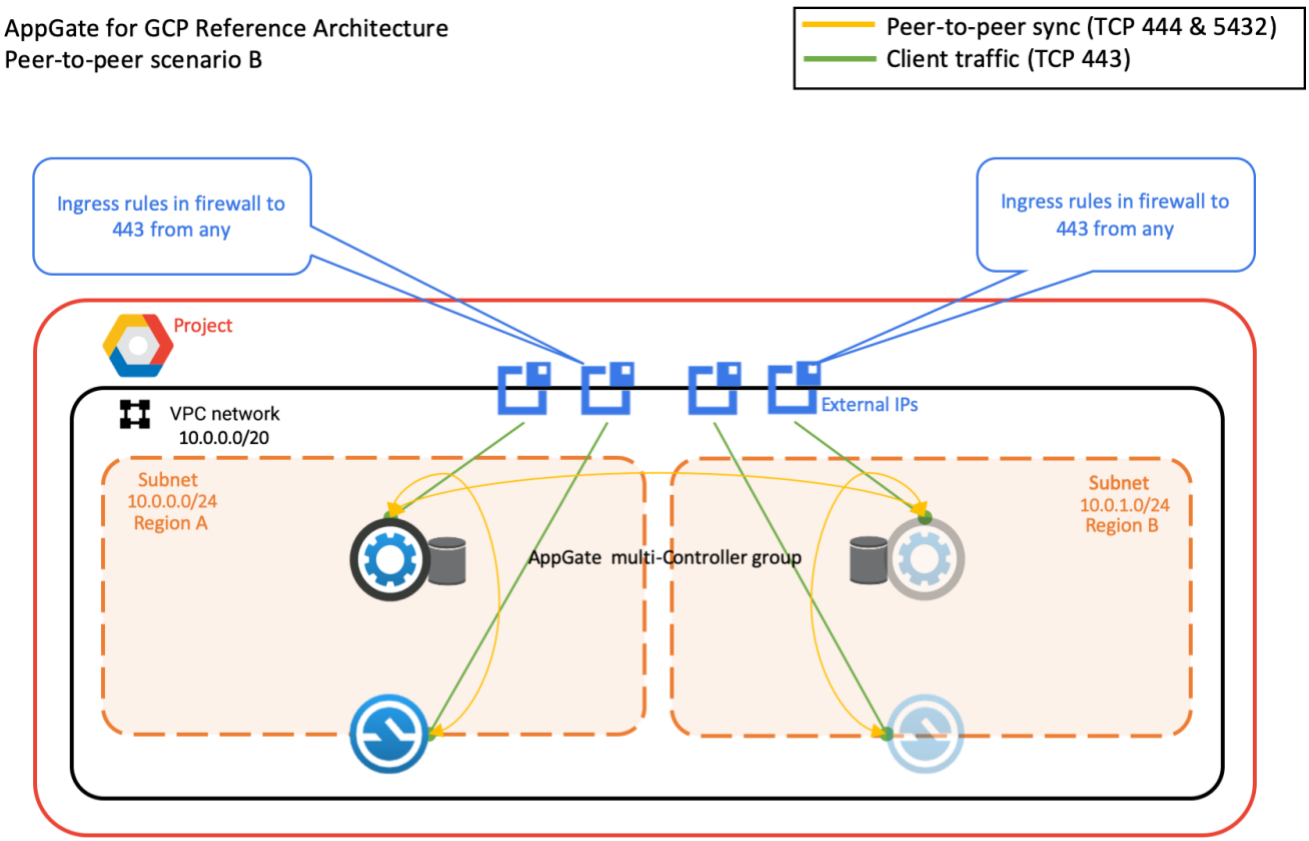


Finally, you then need to regenerate the appliance certificate and the new IP address will appear in the certificate as a **Subject Alternative Names.** Unusually, because we are our own CA, we do allow IP addresses as well as hostnames to be used in certificates.



In this case the external IP address of the appliance is also shown as that is what was set in the Client Interface tab.

This change needs to be made on each appliance. Once done then all the peer-to-peer traffic will now pass within the GCP network.



If the appliances are in different VPCs then care must be taken to ensure that different subnet ranges are used for each appliance. There will also have to be an ingress firewall rule set for 444 (and 5432) for each appliance. This can effectively be automated if an 'Appgate' service account is associated with the Appgate SDP appliances. Then a generalized rule can be set which allows 444 (and 5432) ingress to all VM created using the Appgate service account.

In GCP, even within a subnet, ICMP is not allowed. So, if you plan to use PING to check the internal communications are working then you will need to add an ingress firewall rule set for ICMP for each appliance.

For the sake of simplicity, the scenarios we will all explore later all use external IPs for peer-to-peer communications; but the settings above could be used in all cases.

Controllers

The database is always an integral part of a GCP Controller instance (virtual appliance), however they are shown outside the Controllers just to remind you of the requirement for adequate bi-directional communications (port 5432 is used for database synchronization). The databases rely on eventual consistency — meaning Controller's don't have to wait until all database have processed the record update. There are no session states kept within the database, states are handled by the signed tokens the Controllers send to the Clients. This means that the Controllers only need to store the configuration, Policies and Entitlements, and therefore most database operations will be read operations to generate the secure tokens. The traffic requirements between Controllers is therefore minimal and latency should not affect system performance.

The Controller is normally connected to an Identity Management system, which serves to validate user authentication and act as the source of user attributes and group memberships. The Identity Management system may be located anywhere, as long as the Controller can access it. Appgate SDP supports AD (LDAP), RADIUS and SAML-based identity systems.

Note that in the case of SAML, the Controller acts as the Relying Party, and the Client device authenticates directly with the SAML IDP, relaying the SAML assertions to the Controller. This follows a slightly modified SAML interaction flow. More information about this is available in a SAML guide, available on request.

Gateways

After successful authentication, the Clients obtain the list of Gateways in each pool (called "Site") from the Controller and connect to one of the Gateways in the pool. The Client receives a list of descriptive network Entitlements for each Site. Each network Entitlement token is signed by one of the Controllers so the Gateway will verify the signature before creating a micro private firewall for this specific Client/device combination. The Gateway talks to GCP, to translate descriptive Entitlements like GCP://tag:SSH=Linux-administrators into IP addresses. The micro private firewall will now be configured to allow SSH access to all instances that have this tag within the project. If instances are removed or added with the tag SSH=Linux-administrators, the rules inside the micro private firewall will automatically adapt. Appgate SDP can auto-resolve the instance IP's based on a number of different parameters within GCP. This includes the use of names, tag values and label values.

Clients

After installation, the first time the Client device connects to the Controller the device needs to go through an on-boarding procedure.

Single Packet Authorization (SPA) — requires the Client to open the TCP connection with the Appgate SDP system by using a specific pre-shared key.

The Client requires (to accept) the Controller's certificate as it will be used in this and subsequent communications to verify the authenticity of the Controller. The Client connects on port 443 (TLS) and passes only system control data.

The Client will ask for authentication credentials and optionally a One-Time Password (using the built-in or an external Radius service configured by the customer) to on-board the Client device. The device will receive an on-boarding secure token, which glues the user credentials with the device ID. With on-boarding disabled, a user with valid credentials (username and password) will only be able to use their own on-boarded devices.

The Client generates its own private/public key pair and based on this receives a Client Certificate signed by the CA that will be used for all subsequent mutual TLS connections between the Client and Gateways. This traffic from Clients to Gateway comprises both system control data and application data.

SCENARIOS

Appgate SDP can be deployed into GCP in a number of different ways. The suggested deployment scenarios build on GCP's own reference architectures. However, the GCP reference architectures were not imagined with a Software-Defined Perimeter in mind so the scenarios explored below do depart from the GCP reference architectures in some cases.

Although these scenarios are presented as actual network topologies; the GCP metadata overlays have and equal or in some cases greater effect on the network. For instance, GCP firewall rules are applied by service account or by network tags so by just looking at subnets you only get a partial view of the effective network topology.

GCP has some nice default VPCs and subnets which make getting started very easy. These come with some basic access rules which are applied in the case of these default networks. These typically allow ingress to 22 and 3389 and give the option of ingress to 443. This is only a partial fit with Appgate SDP's requirements. You may need ingress to 443 and/or 444 and at least initially on 22 as well. Other ingress rules may be required such as 5432 when a multi-Controller group is being used. And 3389 is not required as Appgate SDP is based on Linux appliances.

Best security practice would be to only allow 22 and 444 access to the appliances through the Client tunnel. Using an Entitlement Action in Appgate SDP that connects back to localhost.

GCP uses the concepts of projects and overlays this with geographic networking options to ensure high availability. Projects must contain one or more VPC. Each VPC must contain one or more subnets, each of which will be associated with a region. When deploying into a subnet a zone must also be specified. Putting resources in different regions provides a high degree of failure independence (acts of God etc). Putting resources in different zones within a region means the physical infrastructure will be independent (upgrades happen at different times etc).

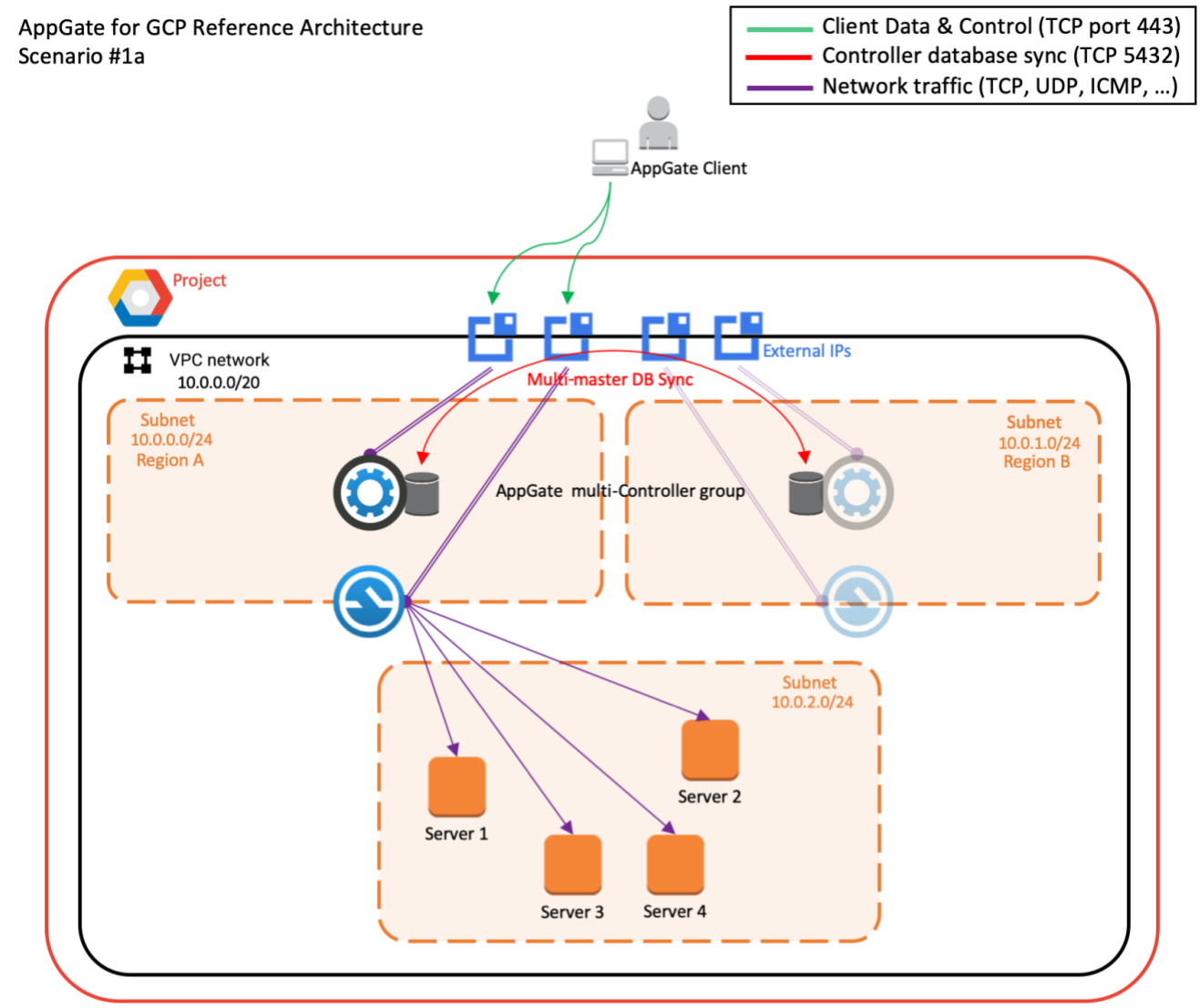
Note that the diagrams use the following icons to represent Appgate SDP components:

Log Server:

Note that the LogServer is omitted from this diagram.

Scenario 1: Appgate SDP in a single GCP project

This scenario the entire Appgate SDP system is deployed within a single GCP project. The protected resources reside in the same VPC as the Appgate appliances.



Architecture Explained

Controller:

The Appgate SDP Controller is deployed as a multi-Controller group split across 2 regions ensuring they are physically separated so that each one shouldn't suffer failures and receive updates at the same time. Whilst the 2 Controllers are shown as being in the same subnets as the Gateways, you could use different ones and locate them in different regions altogether if required.

With 2 Controllers (in the same VPC) normally you might assume peer-to-peer traffic between them would be allowed. In these scenarios the peer interfaces are set to the external IPs so ingress rules will be required for 444 and 5432. The easiest way to do this is to give Controllers a network tag = controller and then allow ingress on 444 and 5432 for VMs based on their network tag. If not already allowed, then also add ingress to 443 for the Controllers (Client access).

Gateway:

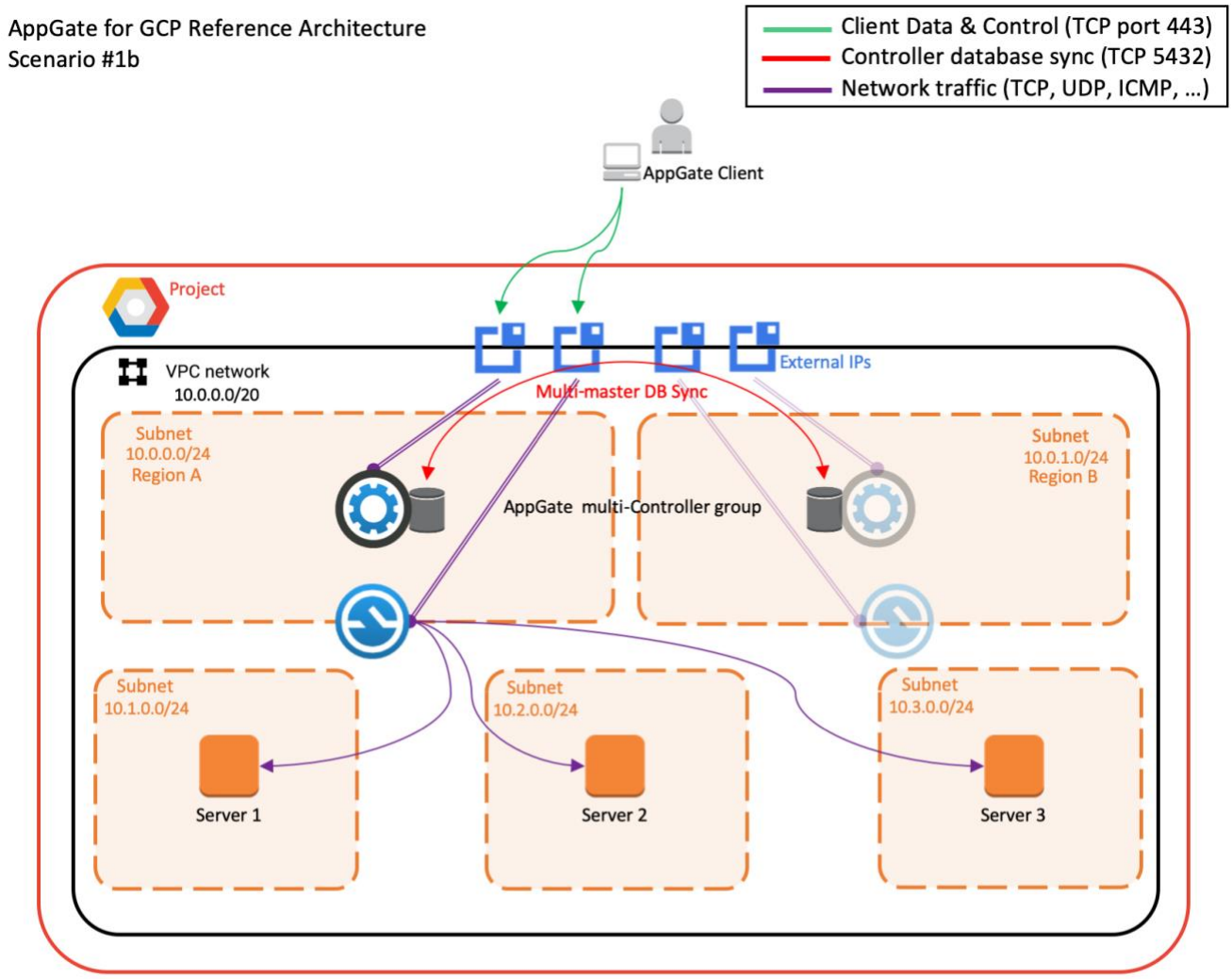
In this example, there are 2 Gateways to provide access to the protected servers in the protected subnet. As with the Controllers, the 2 gateways are split across two regions.

The Gateways are configured with only one interface, sitting in a subnet in front of the protected subnet which contains the protected servers. The Gateways needs to be reachable from outside the GCP network, so each has a GCP external IP assigned to their interface IP address. All traffic between the Appgate SDP Client and Gateways is encrypted.

The Gateways should be set to use SNAT, so that the user's traffic appears to be coming from the Appgate SDP Gateway internal IP address when it is forwarded to the protected servers. For the hosts in the protected sub-net, having the Gateways located in a different sub-net (this is not a specific requirement) is not an issue as all traffic inside a VPC is allowed by default.

The firewall rules for the project will need to allow ingress to 443 and 444 for the Gateways. The easiest way to do this is to give Gateways a network tag = gateway and then allow ingress on 444 (and 443 as well if that is not already allowed). ingress could be blocked for all other traffic to the protected sub-net.

However even in this simplistic scenario you could configure more than one sub-net within a VPC (see Scenario #1b below) and then configure more Sites within Appgate SDP. You could then map the Appgate Sites to the different subnets, possibly using subnet-based routing (SBR).



Benefits and When to Use

This scenario is reflective of a relatively simple cloud environment and is rrecommended by GCP for smaller organizations. This could be utilized in conjunction with a hybrid cloud environment in scenarios whereby the identity provider is cloud-based or in a scenario in which enterprises are migrating to the cloud.

This simple model offers both high availability and Zero Trust enforcement to protected resources. Additionally, with this particular gateway model, users are restricted beyond VPC granularity to specific resources within a subnet, as defined by the Appgate SDP policy.

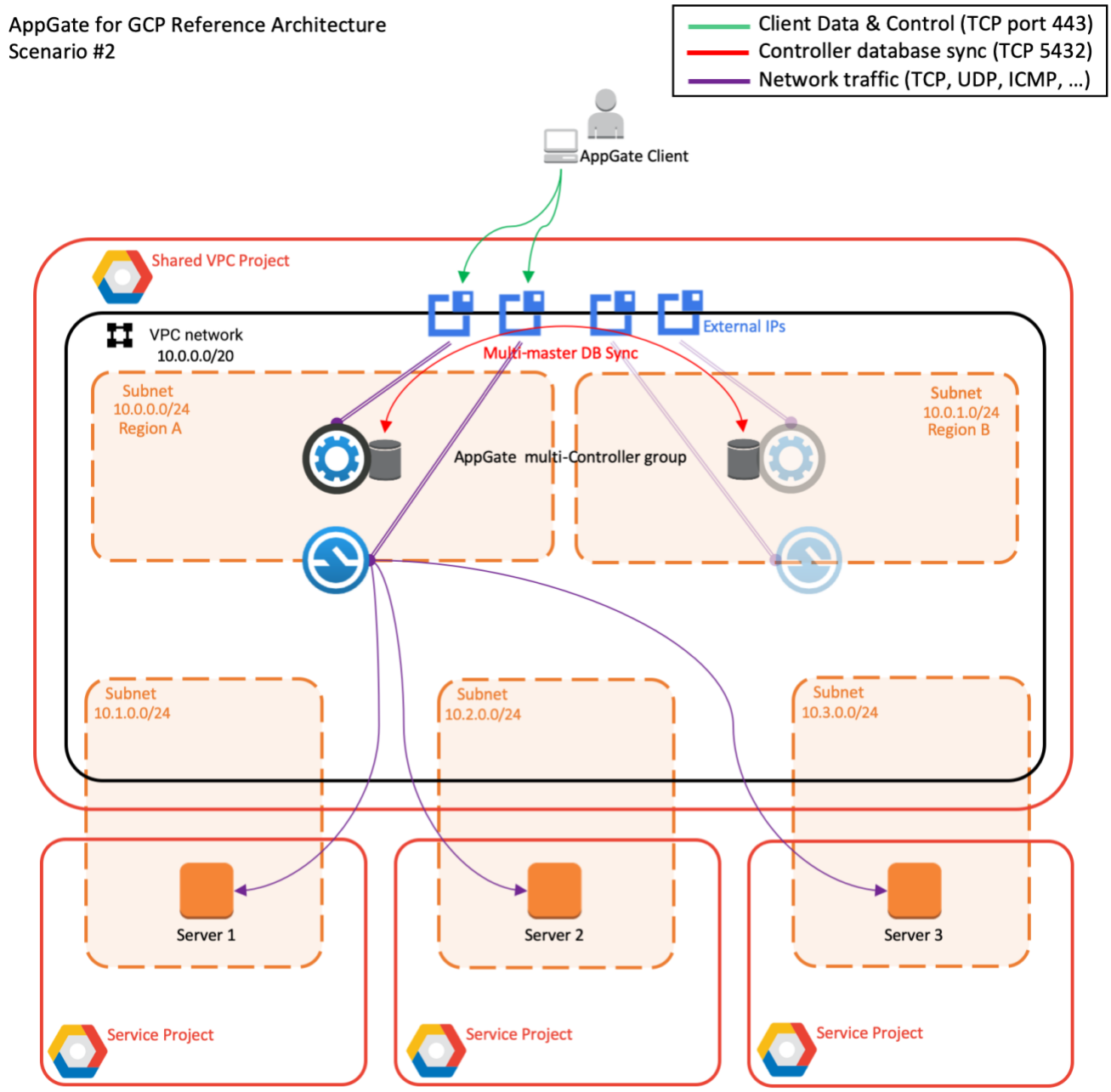
Because GCP's subnets are all region specific, even within one Project and one VPC it is easy to set up ingress points that are local to the users in different regions. The users could be forced connect to the closest of the two Controllers by using some DNS trickery. The Controller could be configured to force the user to the closest Gateway—allowing the long-distance traffic to be routed inside the GCP network.

Scenario 2: Appgate SDP and the Shared VPC

GCP offer organizations with multiple teams the shared VPC. This scenario provides an effective way of extending the architectural simplicity of a single VPC network across multiple working groups. GCP best practice suggests you deploy a single shared VPC host project with a single shared VPC network and then attach service projects for other teams to the Shared VPC host project.

This concept was conceived to allow network policy and control for all networking resources to be centralized. Other departments use service projects to configure and manage non-network resources at a regional level. This is very supportive of say a DevOps model where developers need one or more playgrounds which might come and go over time.

Access to these service projects is from the shared VPC. Since the shared VPC is where the networking is configured, any sort of gateways would normally be located there.



Architecture Explained

Appgate SDP

As far as Appgate SDP is concerned there is no difference between this and Scenario 1b. The only difference here is the existence of the separate service projects. Since these service projects might be ephemeral then it would make sense to use name resolvers to automatically assign the required access rights based on service project meta-data.

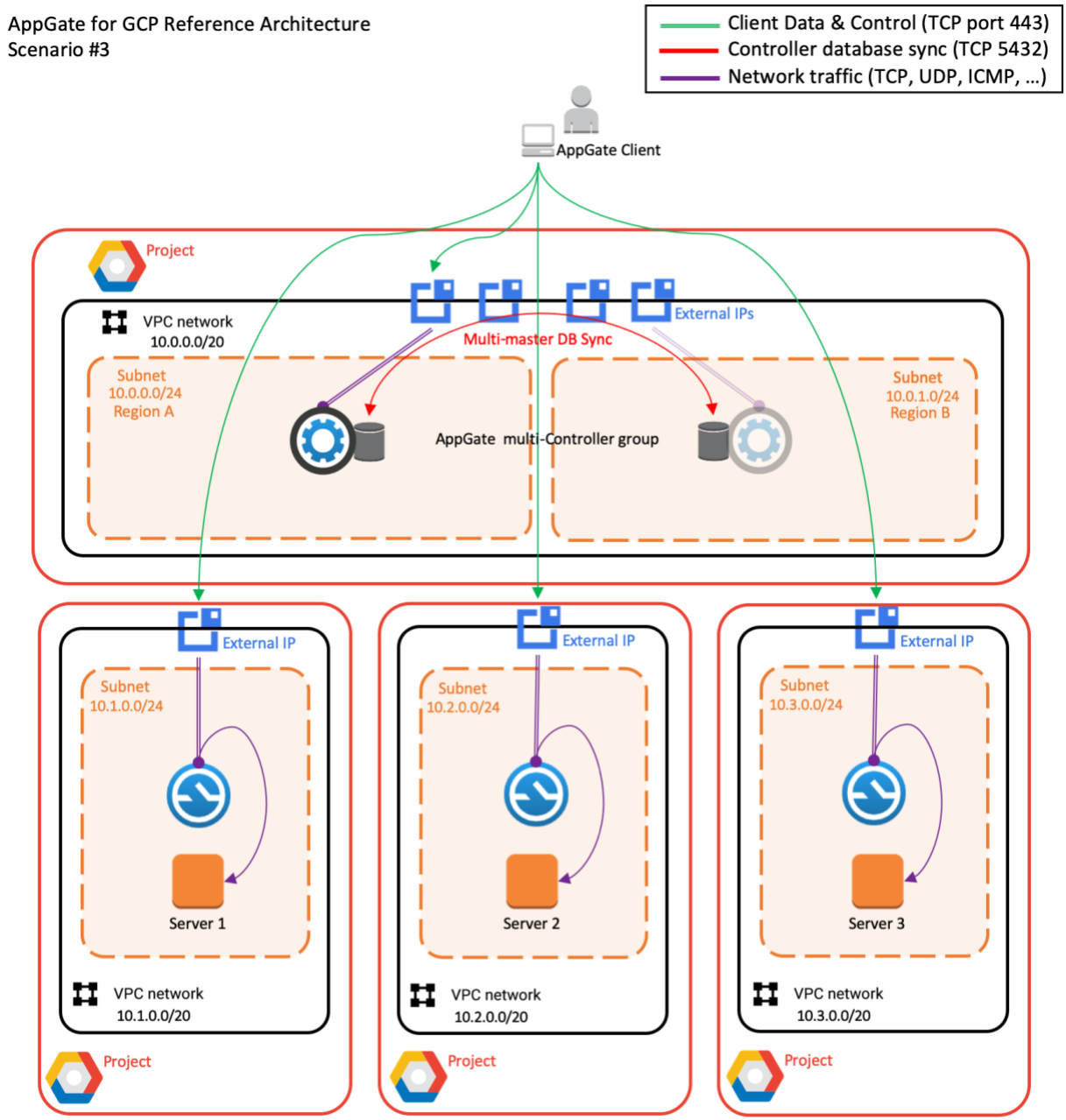
Benefits and When to Use

Appgate provides seamless agility for multi-project environments that utilize different cost structures or billing accounts. This provides users (such as DevOps teams) secure and compliant access without downtime due to waiting for Admins to update IAM roles, firewall rules, etc. Additionally, users can simultaneously access multiple service projects without needing to disconnect.

Scenario 3: Appgate SDP as an alternative to the Shared VPC

The GCP shared VPC is a very good way to manage and simplify the deployment of new projects. Network policy, and control of networking resources and security can be centralized.

This concept does make one big assumption — namely that some key network and security features are best centralized. If there is a dedicated connector to a legacy network or a requirement for a virtualised next-gen firewall then this may be the case. But the point of the Software Defined Perimeter is to distribute this type of functionality, so the adoption of Appgate SDP with its decentralized approach presents an alternative scenario which might better suit some situations.



Architecture Explained

Controller:

As for Scenario 1

Gateway:

In this example we have 3 distributed Gateways dispersed into separate projects/VPCs. Each Gateway provides access to the protected servers in the respective subnets.

The Gateways are configured with only one interface and this sits in the subnet with the protected servers. The Gateways needs to be reachable from outside the GCP network, so each has a GCP external IP assigned to their interface IP address.

The firewall rules for the projects will need to allow ingress to 443 and 444 to the Gateways. Normally ingress is blocked for all other traffic to the sub-net.

For the hosts in the protected sub-net, having the Gateways located in the same sub-net (this is not a specific requirement) keeps the configuration simple.

In this scenario the GCP project would be mapped an Appgate SDP Site (that protects a defined set of servers).

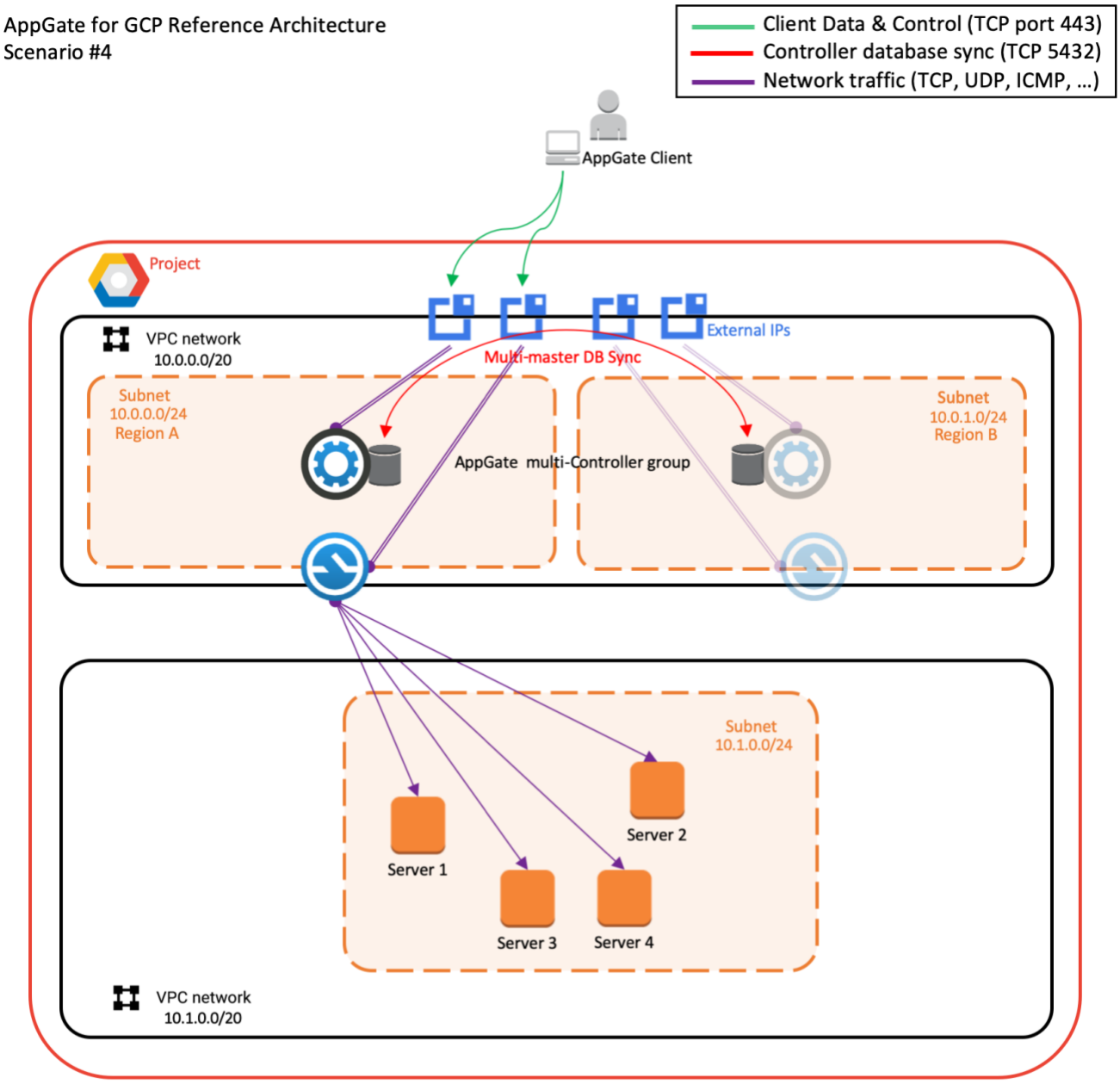
In this scenario only one Gateway is shown per project, if HA was a requirement then two could be deployed.

Benefits and When to Use

This scenario does not rely on having a centralised shared VPC project but still provides many of the advantages of this in terms of ease of provisioning. This scenario is particularly suited to situations where there are very many connections from users/devices coming from outside if the GCP network. It is also well suited to situations where these connections are globally disbursed. Even though the Controllers are shown within GCP, this model would work well in a hybrid environment where the Controllers were still in a company's on-prem data centre.

Scenario 4: Appgate SDP with isolated VPCs

This scenario is likely to be of most interest where you have isolated groups of resources - some of which may require a high level of security such as PCIDSS related servers.



Architecture Explained

Controller:

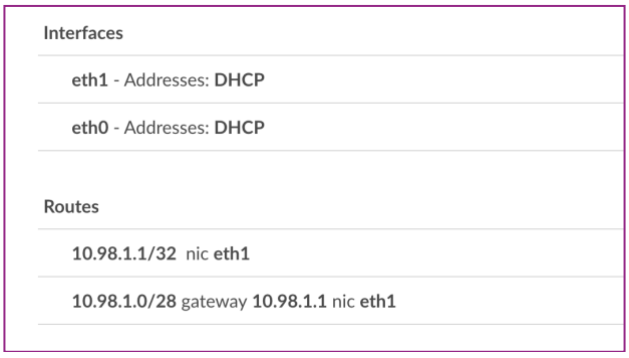
As 1.

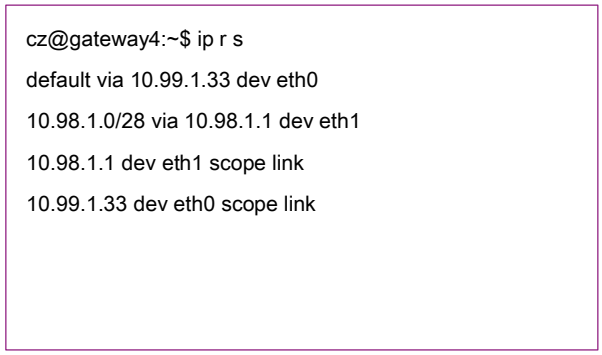
Gateway:

In this scenario, there are still two Gateways providing access to the protected servers. These Gateways are shown as being in the same Project as the protected hosts, but this could equally well be in a separate shared VPC project as in scenario 2.

In GCP it is only possible to define multiple interfaces on VM instances if the interfaces are defined in different VPCs with each in a unique subnet address range. These Gateways are configured with two interfaces and effectively sit between two VPCs. In Appgate SDP DHCP should be enabled on all the interfaces. When using DHCP, Appgate SDP only allows routing information to be used from the DHCP request on the primary interface. This avoids any confusion that would be caused by having multiple default routes defined! When using a single interface this does not matter, however this means the Appgate system will ignore any routing information relating to the second interface. This means there is no information about its local subnet, so a manual route needs to be added in Appgate SDP.

In System>Appliance>Networking>Routes, first add a new route for the Gateway itself — enter the Address/Netmask Length for the gateway which in GCP is always the second IP in the subnet (so for a 10.98.1.0/28 subnet the gateway would be 10.98.1.1) and the Network Interface (eth1) leaving gateway empty. Then add a new route for the local network — enter Address/Netmask Length for the subnet, the gateway (as defined above) and the Interface (eth1). This will now route all traffic destined to the subnet to eth1.





Because the subnets have to be in different VPCs, in GCP this means they are logically isolated from one another. This means the 10.0.0.0/20 VPC can't talk to the 10.1.0.0/20 VPC unless it passes through the Gateway (or there is a firewall rule in place) — which is a perfect arrangement from a security perspective. The Gateways need to be reachable from outside the GCP network so each one has a GCP external IP assigned to their Internet facing interface.

This model could be extended up to the maximum of 8 interfaces (allowed by GCP).

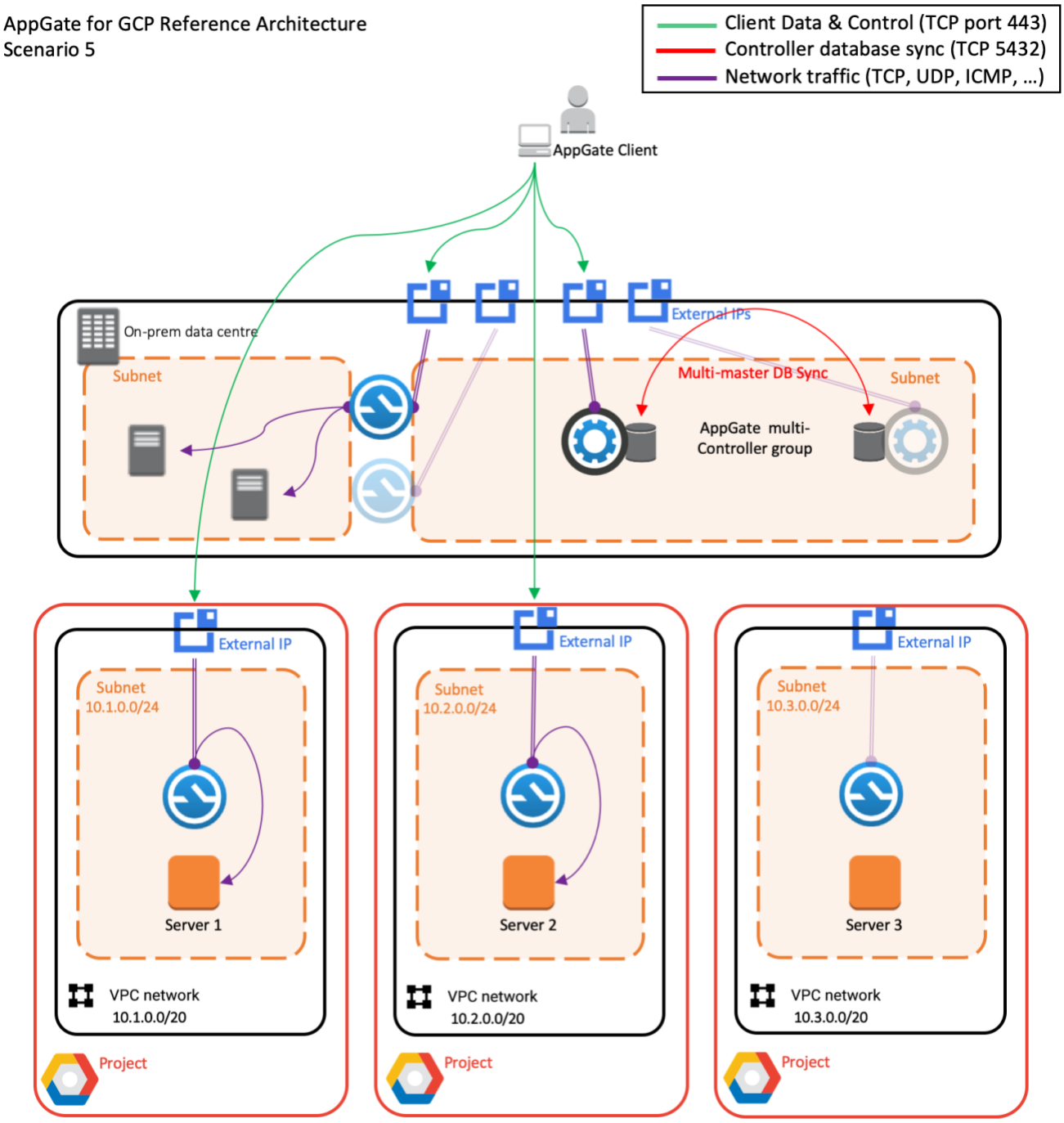
Benefits and When to Use

Again, Appgate SDP could provide seamless connectivity into multi-project or multi-vpc environments (up to 7). This scenario does not rely on having a shared VPC project (as defined by GCP) but does require a VPC to host the Gateways. This model is well suited to a very distributed GCP deployment — possibly happening as part of a ‘services first’ migration to the cloud.

It is also the best option where you want to achieve maximum isolation for one or more VPCs — possibly for compliance related reasons such as PCIDSS

Scenario 5: Appgate SDP Hybrid Deployment

In this scenario, Appgate SDP is deployed in a hybrid model, with the Controllers and Gateways running on-premises, and with 3 Gateways protecting hosts in 3 projects running in GCP.



Architecture Explained

In this deployment model, Client interaction with the Controller is identical to that described in scenario 3. This is a good illustration of Appgate SDP's distributed architecture—showing that Controller location is independent from the Gateways and the protected resources behind the Gateways.

Controller:

With the Controller(s) hosted on premises, it is still possible to assign multiple Sites both on-premises and inside GCP, all managed by the on-premises Controller(s).

Gateway:

The Gateways are deployed the same way as in scenario 3 — each GCP Site is configured with a Gateway to protect the VPC and its respective sub-net.

In this example the Client received Entitlements for an on-premises server, protected by a local Gateway there, and a few Entitlements to access services in the VPC with subnet 10.1.0.0/24 and 10.2.0.0/24 Since the user/device has no Entitlements for the third GCP VPC with subnet 10.3.0.0/24, the Client will not establish a mutual connection to that Site. Even within the allowed VPC subnets the Client only has Entitlements for servers 1 and 2, all others will be invisible to the Client.

Both tunnels are active at the same time and each Gateway will detect if there are any changes within its VPC that requires the micro private firewall within Gateway to update his firewall rule sets. Similar as in scenario 3, the Gateways can be contacted over the internet using an External IP. All traffic to both Gateways will now be encrypted using the Client data channel over port 443 to both Gateways.

Benefits and When to Use

One of the inherent benefits of being "software defined" is that, as software, it is inherently more agile and flexible compared to a physical or even virtualized version of traditional, perimeter-centric security solutions. This illustration shows how the Controller can be resident on a physical premise (perhaps as part of an existing Appgate SDP deployment) and by easily deploying incremental Gateways can provide extend the same security posture to the cloud. Additionally, this model could utilize Appgate SDP's resolvers to dynamically grant on-prem access model to GCP. For example, a user in New York could have access to their local development environment, but only their resources in that environment, their staging environment, and their resources and then the resources they need in the production environment all based on using a common metadata taxonomy.

Instead of users having access to an entire subnet (CIDR block) for staging or production, they now only have access to their specific resources that they need and optimized based on performance, costing and scaling goals (e.g. Dev environment is kept on-prem due to minimize GCP restart costs and source upload time.) Appgate SDP provides this completely seamlessly and out of the box.

RESOURCES

You'll find additional resources on the Appgatewebsite.

The Appgate SDP product documentation is available here:

- Admin Guide: https://sdphelp.appgate.com
- Client User Guide: https://sdphelp.appgate.com/userguide

Access to our support services (including further articles) is via the customer portal.

Thank you, and we hope you find Appgate SDP to be a valuable solution to your security challenges.