appgate

Appgate SDP for Azure Reference Architectures

Type: Technical guide
Date: April 2024
Applies to: v6.3 and newer

Introduction

Some of the largest perceived barriers to migrating workloads and utilizing cloud infrastructure are security and compliance considerations. Azure of course understands this and has created innovative capabilities in security management and regulatory compliance frameworks. They have also partnered with many security solutions providers to extend the security capabilities of cloud based infrastructure.

Appgate is one of those partners and SDP is our leading Software-Defined Perimeter solution available in the marketplace. Appgate provides the following benefits for those deploying in Azure:

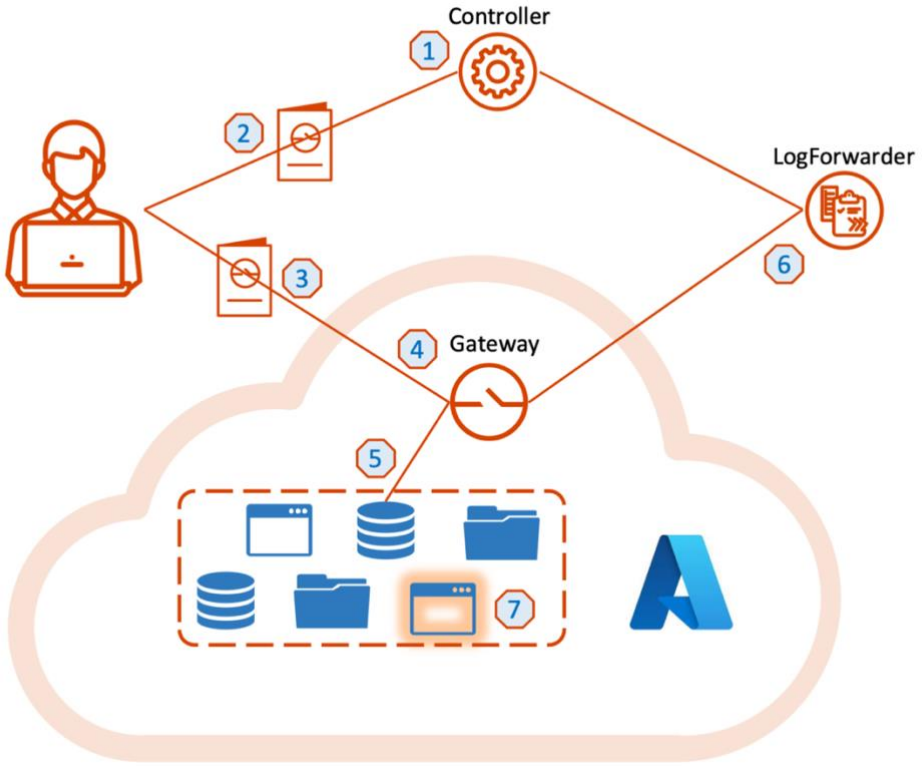
- **Created for the Hybrid Cloud:** Appgate was designed to protect all enterprise cloud resources, including those workloads running in Azure. Appgate has a flexible, distributed deployment model to suit any architecture, automatically detects server instance creation, and leverages user and server attributes to determine access. Appgate also bridges and integrates all elements of the enterprise hybrid cloud infrastructure, controlling access to authenticated users to appropriate cloud resources, wherever they may be.
- **Seamless Integrations:** Appgate can reduce cost, complexity and effort for configuring third-party access, privileged user access and cloud infrastructure management. It combines authorization, encryption and access control in one system, replacing many traditional point products. Appgate also integrates with most identity and SIEM solutions, allowing enterprises to take advantage of existing security infrastructure. This enforces strong authentication, and enables organizations tie network security to their identity management tools.
- **User-Centric Network Security:** Appgate provides application and service-specific authentication and authorization which controls (network) access whether the users are connecting from the inside or outside. Appgate dynamically creates a secure, encrypted network *segment of one* that's tailored for each user session, based on user attributes. Network access rules aren't written once and saved forever, but are created and enforced in real-time.
- **Compliance is Key:** Appgate can help the enterprise reduce regulatory compliance costs by reducing scope and audit complexity. Azure components help with a multitude of regulatory controls, and Appgate can further enhance these. Appgate can also reduce the number systems that fall within audit scope which in turn might eliminate the need for some of the regulatory controls themselves. Robust logging provides all of the possible evidence necessary to meet auditor requirements.

This document provides an introduction to how Appgate might be deployed in Azure, followed by four detailed architecture scenarios.

If you're unfamiliar with the Software-Defined Perimeter (SDP) architecture that Appgate implements, see https://www.appgate.com/resources/ebooks/zero-trust-network-access-everything-you-need-to-know

How does Appgate work (with Azure)?

Appgate utilizes user context to dynamically create a secure, encrypted network ‘segment of one’ tailored for each user session. The Appgate usage model is the same wherever it is deployed and can be described in a few simple steps:



Usage model

- **1. User Authentication:** The user authenticates to the Appgate Controller, which is optionally connected to an enterprise’s IAM system (AD, LDAP, SAML or OIDC)
- **2. Controller Applies Security Policies:** The Controller applies Policies assigned to the user based on user attributes, roles, and context, and then issues a signed Entitlement token listing the resources the user has access to.
- **3. User Accesses Resources:** The authenticated user can now access the SPA protected Gateway.
- **4. Gateway Evaluates User Entitlements:** The Gateway evaluates Entitlements in real-time based on the Access Controls that have been set. For example: network location, time of day, device health, or service metadata, such as security groups. Users may be prompted for additional information, such as a one-time password.
- **5. Gateway Opens Connection to Resource:** If the Gateway determines that required access criteria have been satisfied, it will open a connection to the protected resource specified for the user.
- **6. All Actions are logged:** Throughout the authentication and authorization process, all actions pertaining to the user, resources, and decisions made by the Controller and Gateways are logged. Logs can be stored or sent to the enterprise SIEM for additional action.
- **7. Detects New Services:** The Gateway constantly monitors for the creation of new services that meet certain metadata (such as tags), and any related Entitlements are adjusted as necessary.

SPA

Single Packet Authorization (SPA) keys are used throughout the Collective. The keys used for peer-to-peer connections are generated by the Controller automatically. Clients need an SPA key to be able to connect to the Controller; Client profiles are used for this purpose. Separately to this, keys for Client to Gateway SPA access are generated by the Controller uniquely for each Gateway and are included in the relevant Entitlement tokens. These parts of the SPA process are completely internal and requires no admin or user interactions. For SPA traffic to pass UDP ports 53 & 443 and TCP ports 443 should all be open.

Controllers

The Controllers utilize multi-master database synchronization for high availability, based on an eventual consistency concept. Controllers therefore require bi-directional connectivity which is used for messaging and mutual database synchronization.

The reason why eventual consistency can be used is that there is no session state kept within the database, the state is handled purely by the signed tokens the Controllers send to the Clients. This means that the Controllers only need to store (static) information such as settings, Policies and Entitlements. Most database operations will be read operations as part of the process to generate the secure tokens.

Clients connect using the built in DNS round-robin HA model. (Although not recommended, the Azure Global Load balancer or the Azure Traffic Manager can be used to distribute the traffic based on Performance, Weighting, Priority or Geography.)

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(s) can access it. Appgate supports AD, LDAP, RADIUS, SAML and OATH 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, Clients obtain the list of Gateways for each “Site” from the Controller as well as a list of descriptive network entitlements for each Site. The Client then connects to one of the Gateways in the Site using mutual SPA protected TLS traffic on port 443.

Each Entitlement token is signed by the Controller; the Gateway verifies this signature before creating a micro-firewall instance for this specific Client/device combination.

The Gateway also talks to Azure via APIs, to translate any descriptive Entitlements like `azure://{"subnets":"Linux"}` into IP addresses. The micro-firewall will now be configured to allow access to all instances in the Linux subnet within the Resource Group. If instances are removed or added to the subnet, the rules inside the micro-firewall will automatically adapt. Appgate can auto-resolve the instance IP’s based on a number of different parameters within Azure. This includes the use of names, tag key or tag value referenced against hosts, Networks Security Groups, Virtual Networks, SubNets and Load Balancers.

Gateways do sometimes need to connect to Controllers, however this is not required when a Client connects as all the information required is already in the Tokens.

There is no communication between Gateways whatsoever.

Clients

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

Single Packet Authorization (SPA) is used to open the TCP connection to the Appgate Collective. The Client checks the Controllers certificate against the fingerprint as it will be used in this and subsequent communications with the Controller. The Client typically only exchanges control data and logs with the Controller.

The Client will ask the user for authentication credentials and optionally a One Time Password (using the built-in or an external Radius service configured by the customer) to register the device. The device will receive secure token tied to that device's ID. It is possible to configure the Collective to only allow the use of already registered devices if required.

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 Gateway TLS connections. The traffic from Clients to Gateway comprises both control data and application data.

Scenarios

Appgate can be deployed into Azure in a number of different ways. Below are some example deployment scenarios, with explanation and analysis.

When using Network Security Groups in these scenarios then the default settings might only allow 22 and 443 inbound to any given Appgate appliance. So, you may need to remove 22 and even open 8443 (admin access to Controllers and LogServers). Best security practice would be to only allow port 22 access to the appliances through the Client tunnel. Effectively using a firewall rule in Appgate that connects over the local (protected) network to port 22.

Even though one or two 'catch-all' Resource Groups are shown in these scenarios, these don't really affect the underlying architectures so for the most part these can be ignored.

Controller

Appgate Controllers are deployed wherever they are required. This might for instance be using Appgate's own ZTP hosted service. Whilst the 2 Controllers are shown together there is no reason why they need to be co-located – indeed it might be good practice to have a Controller located close to the user community to reduce sign-in latency.

LogForwarder

The LogServer does not form part of the scenarios that follow.

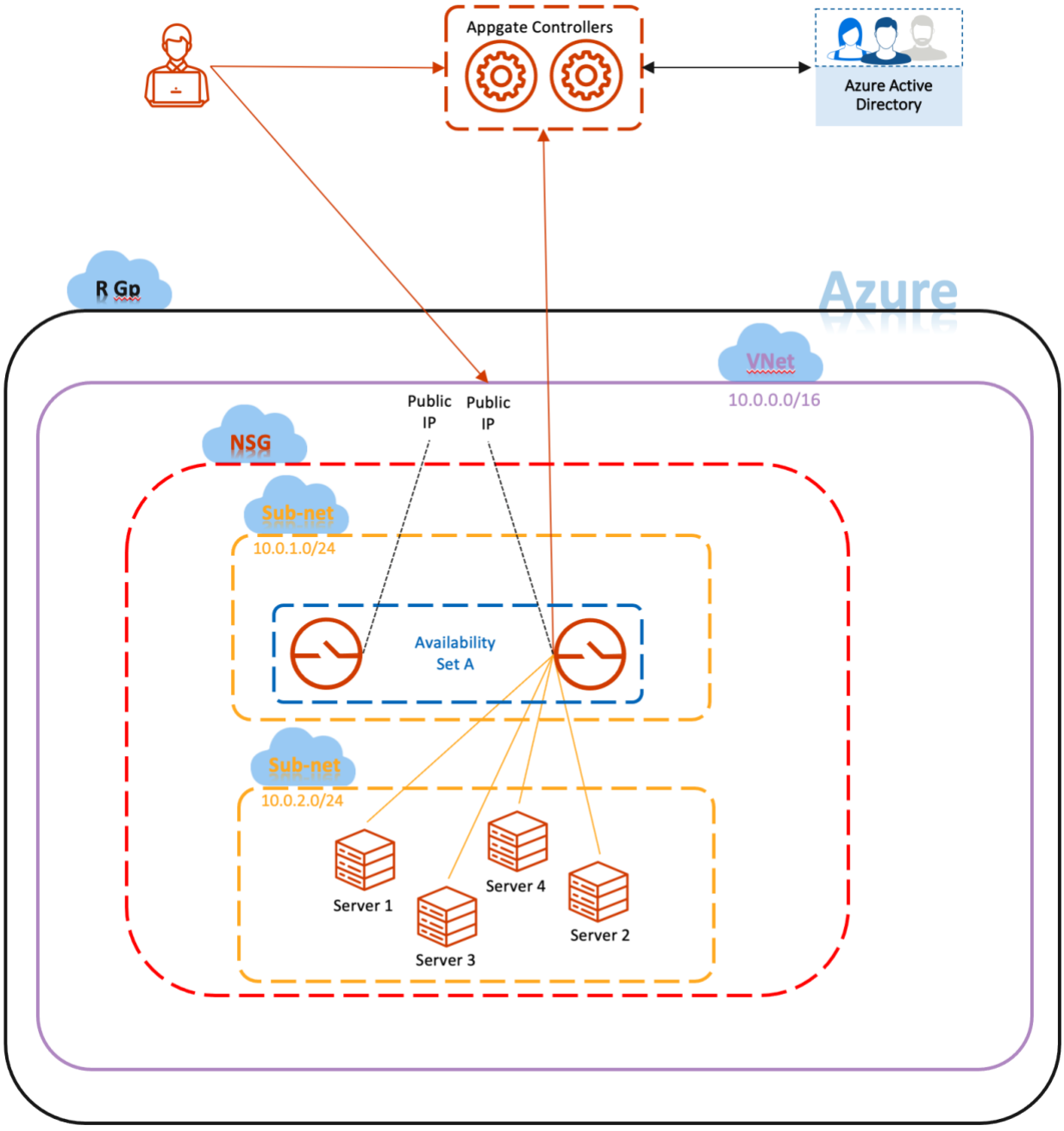
Note that the diagrams use the following icons to represent Appgate components:

Controller

Gateway

Scenario 1a: Appgate in Azure protecting single location

In this scenario, there is just one Site defined - which sits within Azure. The protected resources sit together in one VNet. The Azure Availability set settings can be used to ensure that at least one of the Gateways are available at all times to provide access to the protected servers.



Scenario 1a - Details

In this example, there is just one Appgate Site defined - which sits within Azure. there are 2 Gateways to provide access to the protected servers in the 10.0.2.0/24 sub-net.

The Gateways are configured with only one interface; they sit in their own sub-net and are in front of the protected (Azure) sub-net which contains the protected servers. The Gateways needs to be reachable by Clients (outside the Azure network) so each has an Azure Public IP assigned to their interface IP address.

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

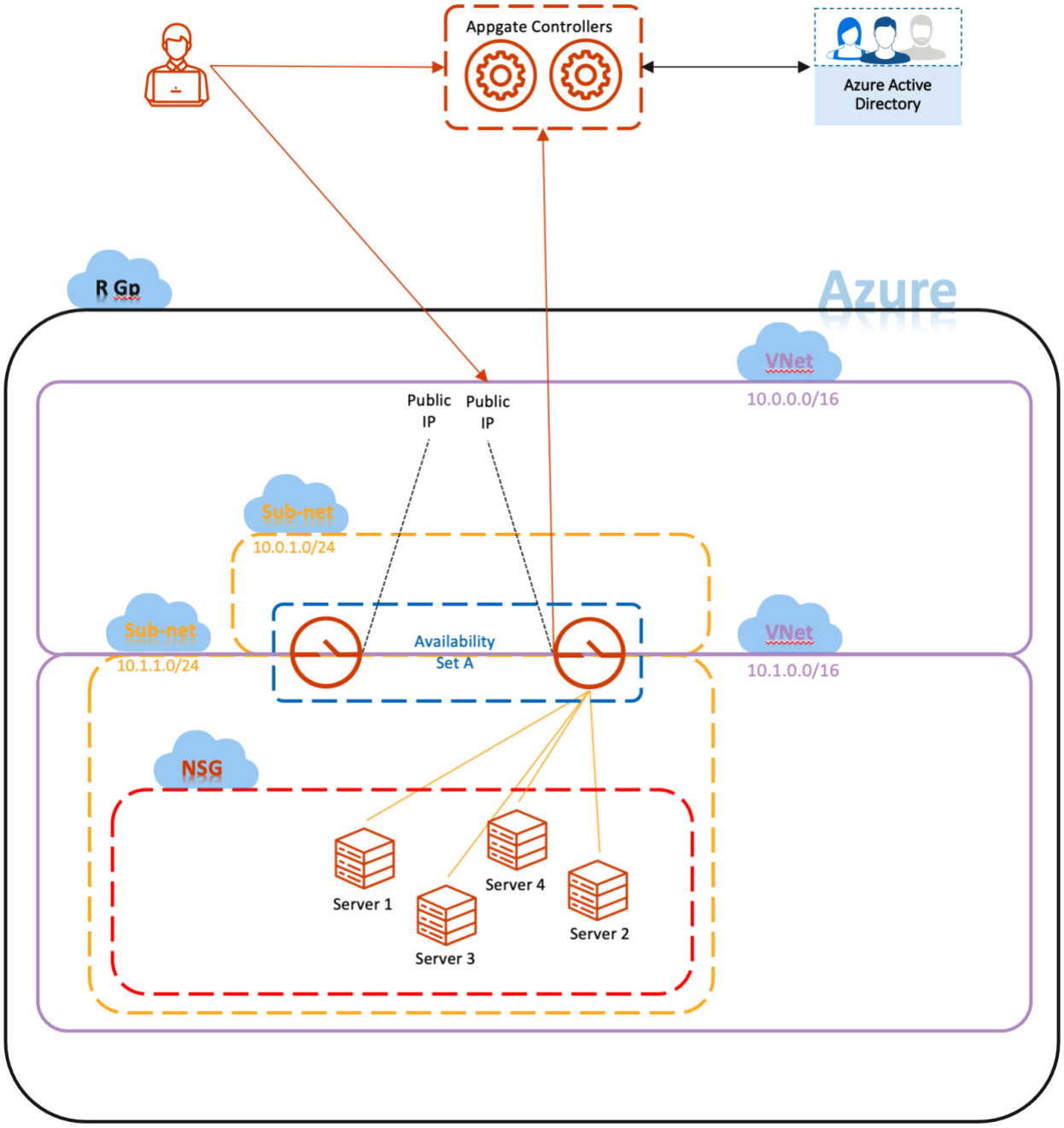
Most organizations will probably wish to configure the Network Security Group (NSG) for the hosts in the protected sub-net. The use of a separate sub-net will make this easier to configure. The NSG rules can be set for the subnet to only allow inbound access from the Gateway's IP address (and might also block all outbound internet traffic if the protected hosts don't require this).

The NSG could also be used to limit inbound/outbound traffic to the Appgate SDP sub-net. This should allow 53 UDP and 443 UDP/TCP inbound. Gateways themselves make only outbound connections to other appliances in the Collective so this will typically work with the default rulesets (but should allow 53 UDP and 443 UDP/TCP).

When using Azure it is probably easiest to map the Appgate Site (that protects a defined set of servers), to Azure sub-net(s). You can configure more than one sub-net within a VNet and more Sites within Appgate. This can extend to different VNets and even different locations (as shown in one of the next examples).

Scenario 1b: Appgate in Azure protecting single location

In this scenario, there is still just one Site defined - which sits within Azure. The protected resources sit in a different VNet. The Gateways have 2 interfaces configured – one, internet facing and the other, talking to the protected servers.



Scenario 1b - Details

In this example, there are 2 Gateways to provide access to the protected servers in the 10.1.1.0/24 sub-net.

In Azure it is possible to define multiple interfaces on Gateway instances and assign these to different sub-nets. This must be done when the instance is created and not afterwards. In this scenario, the Gateways are configured with two interfaces, so it effectively it sits between two sub-nets. If the instances have limited throughput quotas for the NICs then this arrangement can be good if more throughput is required.

These sub-nets can be in different VNets or even Resource Groups. In this case the sub-nets are in different VNets (which in Azure are logically isolated from one another). This means nothing in the 10.0.1.0/24 sub-net can talk to the 10.1.1.0/24 network unless it passed through the Gateway server – which is a perfect arrangement.

The Gateways needs to be reachable from outside the Azure network so each has an Azure Public IP assigned to their 10.0.1.x internal interface.

Most organizations will probably wish to configure the Network Security Group (NSG) for the hosts in the protected sub-net. The use of a separate sub-net will make this easier to configure. The NSG rules can be set for the subnet to only allow inbound access from the Gateway's IP address (and might also block all outbound internet traffic if the protected hosts don't require this).

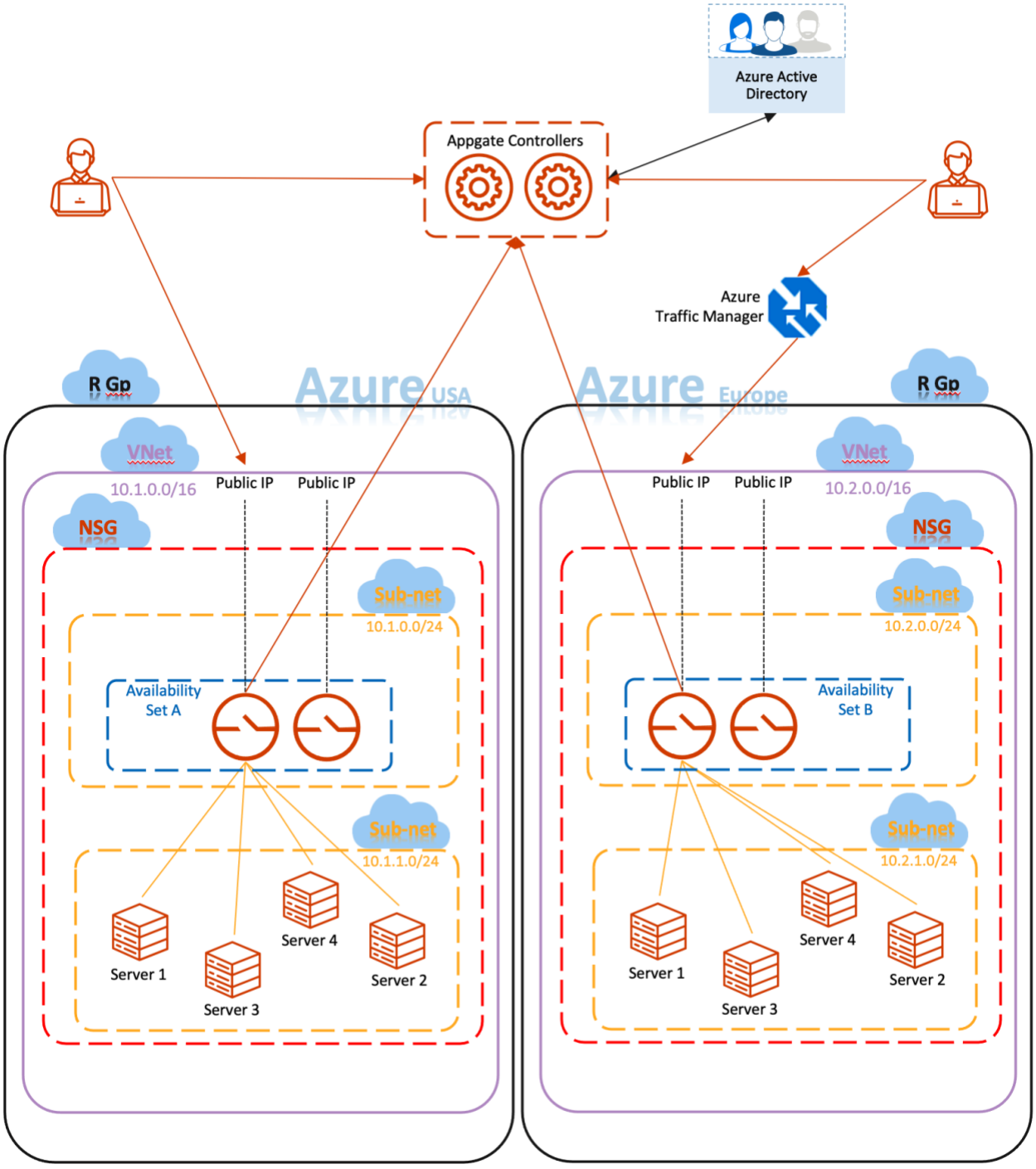
The Appgate Gateways already include a firewall (IP Tables) so traffic to the appliances from the Internet will already be limited to 53 UDP and 443 UDP/TCP inbound. Another NSG could be used to further limit inbound traffic to the Appgate SDP sub-net if required.

Gateway will detect if there are any changes within its VNet that requires the micro private firewall within Gateway to update his firewall rule sets.

Scenario 2: Appgate in Azure – geolocated user access

This is similar to scenarios covered in (1) however, in this case there are two (or more) Sites, each protected by Gateways. These Sites are effectively replicas of one another – the user only needs to connect to one of them. This would typically be done to provide the best user experience with locally accessed protected resources.

When you create a Resource Group in Azure you define the Region at that time. In this example the two Azure Resource Groups have been set up in the USA and Europe. You can now do the same in Appgate when you define the Site you have the option to enter the Geolocation.



Scenario 2 - Details

In this scenario, we want to have users connect to the closest of the two Sites by default.

The Controllers could also be split up and geolocated if required – however the sign in process requires only a very limited amount of traffic, which means there is seldom any real benefit to doing this. Additionally, there are usually several time hungry steps (such as querying LDAP) which would outweigh any potential latency gains.

Having said that Traffic Manager could be used in this case but is not recommended as some of the HA features of the Controller will be lost (see HA for more details).

The Gateways are in different Azure Resource Groups and Regions – each Site is configured with redundant Gateways to protect the VNet and its respective sub-net. These would both be placed in the same AS to ensure one was always available.

In this case the requirement is for the left user to connect to a Gateway running in the Azure USA resource group, and have its tunneled traffic directed to the servers running in that VNet.

Azure offers several services to support this way of working:

- Azure Traffic Manager operates at the DNS layer resolving incoming DNS requests based rules (including geolocation). This method determines which public IP to connect to. For this to work, the DNS name of the Traffic Manager has been given to the right Client and because the Traffic Manager is monitoring the Gateways' healthcheck service it knows which Gateways are operational.
- Azure cross region load balancer has a global IP address and forwards traffic to the closest Azure region. Again it should monitor the Gateway healthcheck service to ensure the Gateway is operational.
- Azure Load Balancer can't be used across regions.

Using Azure geolocation services is not ideal as the Appgate Client is designed to connect to all allowed Sites – so the two regions would have to be defined as being on the same (Appgate) Site! The geolocation service would therefore need to know the Client hostname/IP for each of the 4 Gateways.

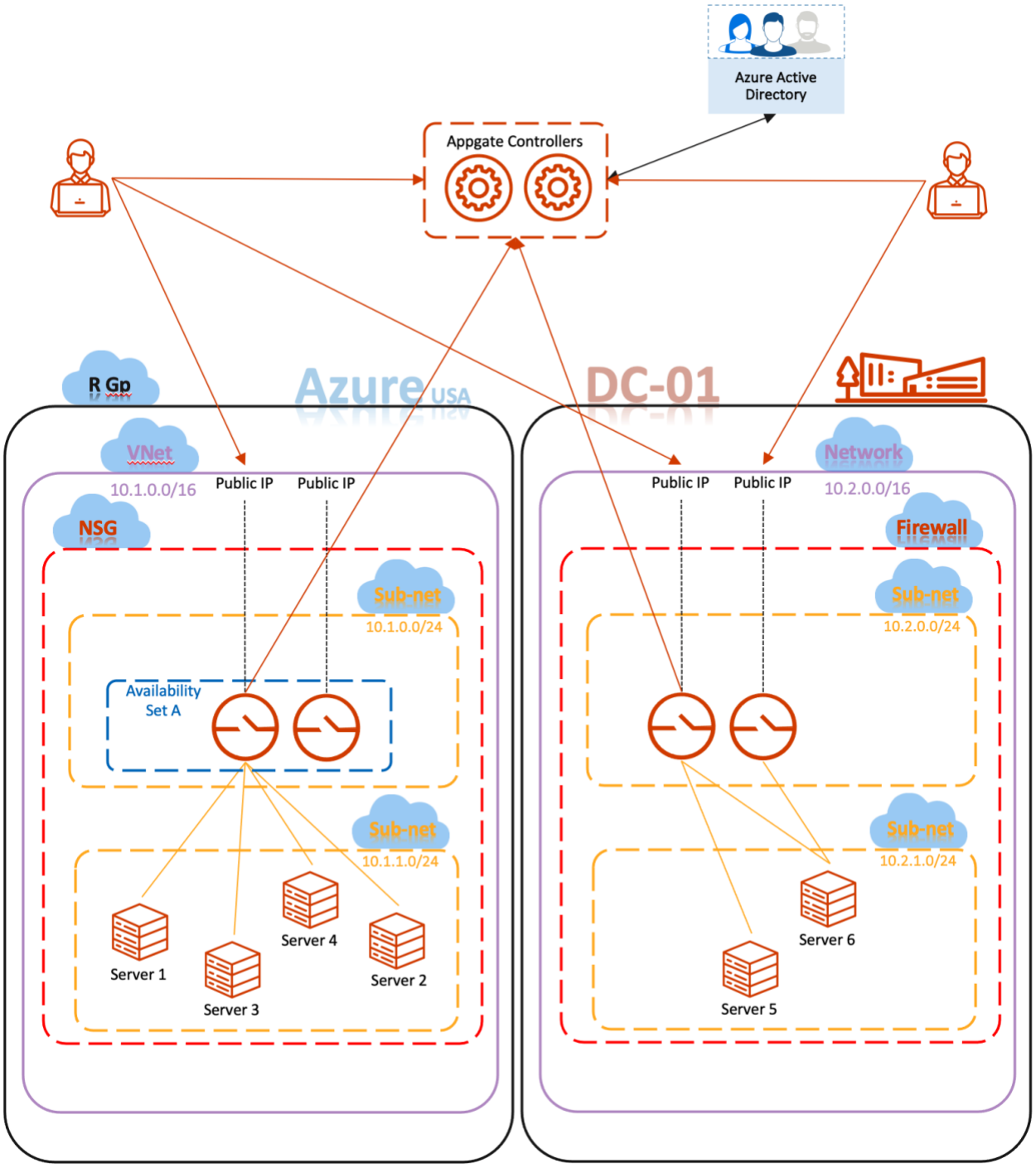
Instead of the above scenarios, it is better to use the built-in mechanism providing geolocated access to Sites. The two Sites can then be defined separately in the two regions. Each Site is set up with a geolocation and with *Use for nearest Site selection* enabled.

The Policy can then be set up with *Override with the nearest Site* and when the user signs-in, the Client's geo claim is used to select the nearest allowed Site. Because the default Site is 'overridden' the Client is given an Entitlement to just one (the nearest) of the two geo-enabled Sites.

In general it not recommended to use (Azure's) geolocation services for this as some important built-in HA features of the Gateways will be lost (see HA for more details).

Scenario 3: Appgate in Azure – protecting distributed resources

In this scenario, Appgate is deployed in a hybrid model. The Controllers could be cloud hosted as part of Appgate's ZTP service offering. Gateways might take the form of physical appliances running in a private data centre, and Azure deployed Gateways protecting hosts located in Azure.



Scenario 3 - Details

In this deployment model, Client interaction with the Controller is identical to that described in scenario 1. This is a good illustration of Appgate's distributed architecture – showing that Controller and Gateway locations can be set to meet the requirements of the business irrespective of where the protected resources are located.

The Azure Gateways are deployed the same way as in scenario 2 with redundant Gateways to protect the VNet and its respective sub-net.

The data center's Gateways can be physical or virtual and again redundant Gateways are specified to protect the VNet and its respective sub-net.

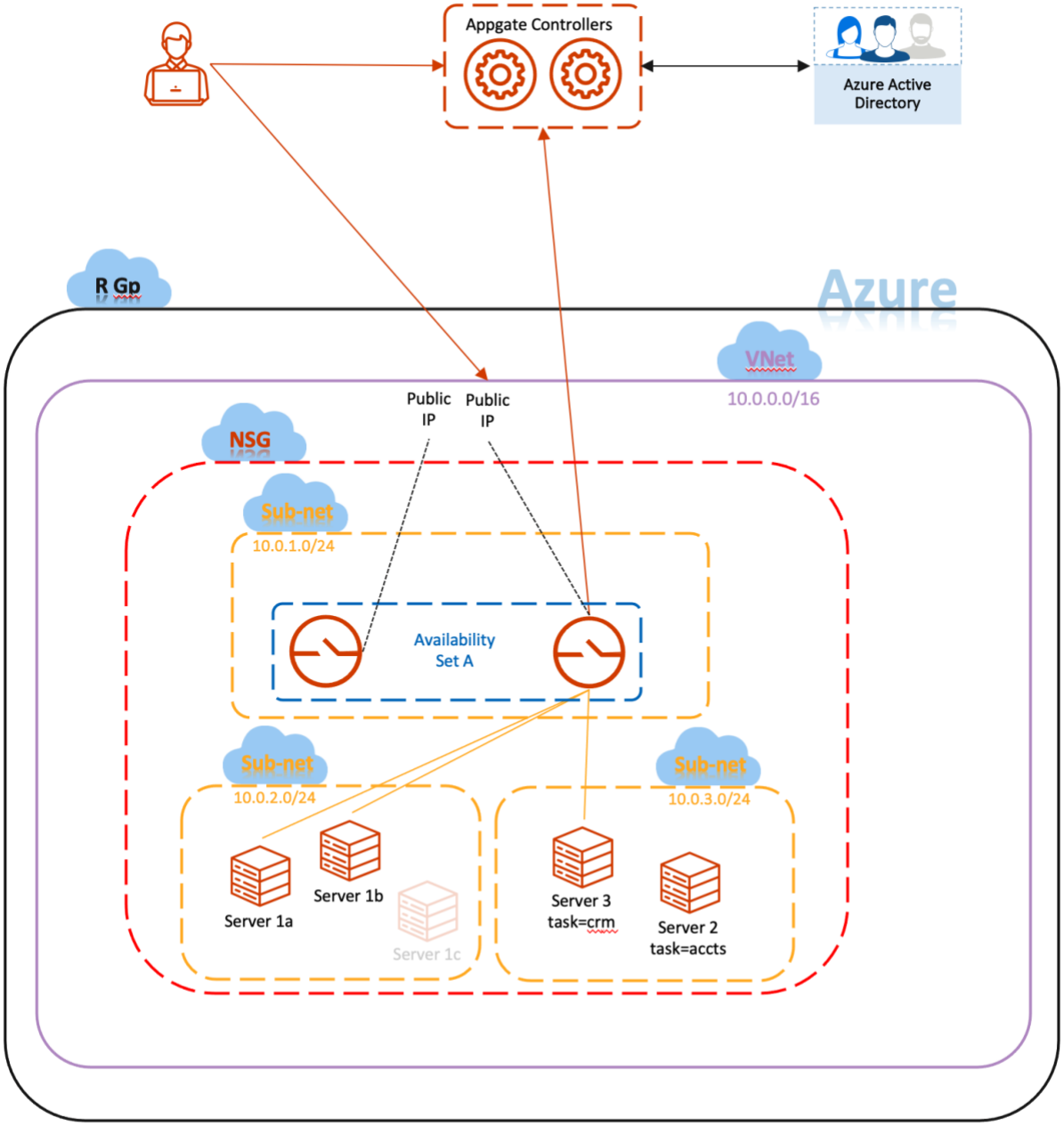
In this scenario the left Client received Entitlements to access instances in the Azure VNet as well as Entitlements for some DC based servers. Both tunnels are active at the same time and the user experience will appear identical for both Sites.

The right Client received an Entitlement to access just one DC based server. The right Client doesn't have Entitlements for Azure and the Site and instances will be invisible to this Client.

This scenario is much closer to the true SDP model where Clients connect to all the allowed Sites.

Scenario 4: Appgate in Azure – using resource names to define hosts

This is similar to scenarios covered in (1) however, in this scenario there are two sub-nets, both protected by Gateways. The left subnet contains Server 1 which is set to autoscale. The right subnet contains two more non-autoscaling servers.



Scenario 4 Details

In this deployment model, Client interaction with the Controller is identical to that described in scenario 1. However, the way the Controllers construct and issue Entitlements is very different.

The Gateways are deployed in a similar way to scenario 1 – The Azure Site is configured with redundant Gateways to protect the VNet and its respective sub-nets.

In this example the left subnet is reserved for a specific auto-scaling instance. 1a and 1b are active 1c is not active at this time. The right subnet has two instances which have had tags assigned to them (that describe their role).

Resource names have been used to define what access rights the user will be granted. Instead of defining access using IP addresses, the syntax would look like this:

```
azure://{"subnets":"auto"}
```

The name resolver in the Gateway will then query Azure for all the IPs in the subnet “auto” which will then be added to the user’s micro-firewall rules.

```
azure://{"virtual-machines-tags":{"tag-key":"task", "tag-value":"crm"}}
```

The name resolver in the Gateway will then query Azure for all the IPs belonging to instances tagged with “task:crm” which will then be added to the users micro-firewall rules.

The user’s Entitlement for this Site will contain 3 instances (2 from the subnet and 1 that was tagged). The name resolver queries Azure every 60s so when 1c becomes active the new instance will be automagically added to the users Entitlement. Similarly, when 1c is deactivated then it will be removed from the Entitlement again.

Resources

You’ll find additional resources on the Appgate website

There is another step-by-step guide relating to the templated apps in the Azure marketplace.

There is a specific guide relating to Using name resolvers in Azure

The Appgate SDP product documentation is available here:

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

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