**Initial Setup:**

1. Setting up network rules :
   1. Obtain your ISM Gateway IP:
      1. Access the Internal Scanning Management admin panel:
         1. A screenshot of a computer

            AI-generated content may be incorrect.
      2. Save the IP address listed next to the Gateway Status.
   2. Allow for Outbound access to your ISM Gateway IP:
      1. Make sure it’s set to TCP and port 443.
   3. Allow for Outbound access to your application.
   4. Allow the ISM to connect to your application/service.
      1. This will depend on how the application/service is deployed.
2. Building a container to add to your container registry:
   1. Download the latest version of the Linux ISM installer.
   2. On a system with Docker installed, download the repository located at <https://github.com/cadonuno/veracode-ism-container/> and put the ISM installer on the root of the project.
   3. In Veracode, create a new endpoint (or, for an existing one, select Actions->Set Up).
      1. Save the Endpoint Key listed at the bottom.
   4. Open a command prompt at the folder containing the Veracode-ism-container project and run the following commands (replacing anything in bold):
      1. docker build -t veracode-ism-container:**<ism-version> .**
         1. This will set up the container.
      2. docker run Veracode-ism-container:**<ism-version> --token <your-ism-token>**
         1. This will print your ISM key, save it as we will need it later.
      3. docker tag Veracode-ism-container: **<ism-version>**

**Azure Container Registry:**

1. Add the container to your Azure container registry:
   1. Under the Azure Container Registries configuration, open the Access Keys menu:
      1. From there you will need:
         1. **Login Server.**
         2. **Username.**
         3. **Password.**
   2. Open a command prompt at the folder containing the Veracode-ism-container project and run the following commands (replacing anything in bold):
   3. docker login -u **<your Azure Container Registry Username>** -p **<your Azure Container Registry Password>** **<your Azure Container Registry Login Server>**
   4. docker push **<your Azure Container Registry Login Server>/**Veracode-ism-container:**<ism-version>**
2. Create the Azure container:
   1. Select the Create Container instance Option.
   2. Under Basics set:
      1. Resource group.
      2. Container name.
      3. Image Source: Azure Container Registry
      4. Image : veracode-ism-container
      5. Image tag: latest
      6. Size (important):
         1. **Click the Change size button and set the Memory to at least 8 GiB.**
      7. Under Networking set:
         1. Networking type: private
      8. Virtual network: the one we created (or edited) above.
   3. Under Advanced:
      1. Set the Restart policy to Never
         1. We will be launching this container on demand.
      2. Create 2 environment variables, both set as secure **Yes**:
         1. **TOKEN:** Set this to your ism token
         2. **APIKEY:** Set this to the ism key returned when running docker run earlier.
      3. Set the command override to **["/startup.sh"]**
   4. Review + Create and save the container instance.

**AWS Container Instance:**

For AWS, we can obtain the container push command in AWS directly, however, we need to install the AWS CLI (<https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>).

1. Create 2 secrets in your Secrets Manager:
   1. **VERACODE\_ISM\_TOKEN:** Set this to your ism token
   2. **VERACODE\_ISM\_APIKEY:** Set this to the ism key returned when running docker run earlier.
2. Under ECR create a new repository.
3. Inside the repository, click ‘view push commands’
4. Run the commands provided from a CLI with docker and the Amazon CLI.
5. Under ECS, create a cluster.
6. Create a task definition:
   1. Make sure to:
      1. Under Task Size, set memory to 8GB.
      2. Under Container, set it to your new AWS container image.
      3. Under Environment Variables, using valueFrom, link these 2 values:
         1. **TOKEN:** the secret containing your ism token
         2. **APIKEY:** the secret containing your ism key returned when running docker run earlier.
7. Create a new Service under the cluster and set its task family to the new task definition
   1. Make sure it is set not to relaunch on failure.