SYNTASA

www.syntasa.com

SYNTASA Infrastructure Setup for Air-Gapped Environments

SYNTASA 6.2+ Infrastructure setup document for Air-Gapped environments running in AWS Cloud.

Prepared by SYNTASA

Prepared on 03/07/2022

Table of Contents

1. INTRODUCTION 3
1.1 PURPOSE 3
1.2 DOCUMENT INFORMATION 3
2. AMAZON CLOUD OVERVIEW 4
3. SYNTASA AWS ARCHITECTURE 5
AWS Architecture overview 5
4. AMAZON S3 BUCKETS 6
5. AMAZON POLICIES AND IAM ROLES 7
S3 Access Policy 7
Kubernetes Node Role 7
EMR Default Role 8
EMR EC2 Default Role 8
EMR Autoscaling Default Role 8
6. AMAZON VPC AND NETWORKS 9
VPC Setup and Configuration 9
RANCHER Worker Subnet 1 9
RANCHER Worker Subnet 2 9
RDS Instances Subnet 9
EMR Cluster Subnet 10
SUBNET Route Tables 10
7. AMAZON SECURITY GROUPS 11
RANCHER Worker Security Group 11
RDS Security Group 11
EMR Master Node Security Group 12
EMR Worker Node Security Group 12
EMR Service Node Security Group 12
Load Balancer Security Group 13
8. AMAZON RDS METASTORES 14

Syntasa Application Metastore 14
Syntasa Application Metastore 14
9. ADDITIONAL SETUP ITEMS 15
EC2 Key Pair *15*
Installation Server Instance *15*
Kubernetes RKE Cluster *15*
Kubernetes RKE Node Pools and Sizes *16*
Kubernetes RKE Node Pools and Sizes *16*
INFRASTRUCTURE SETUP COMPLETE 18

1. INTRODUCTION

1.1 PURPOSE

This document contains the installation and setup of AWS Cloud Resources steps needed for the installation and configuration of the SYNTASA platform in Air-Gapped environments.

1.2 DOCUMENT INFORMATION

Produced Date: 03/07/2022
Prepared By: SYNTASA DevOps
Produced For: SYNTASA Customers / General Release
Revisions: N/A

ORIGINAL DOCUMENT PREPARED ON – March 7th, 2022

a.) No Additions

2. AMAZON CLOUD OVERVIEW

This document outlines the infrastructure pieces needed to successfully setup the SYNTASA Data Analytics and Machine Learning platform in the Amazon Web Services Cloud environment. This document assumes that the client AWS project is set up and the requisite permissions configured to install and setup the following Cloud Services:

AWS Cloud Service Name
VPC | Subnets | Route Tables | Security Groups
RDS Instances | S3 Buckets
IAM Roles | Policies | Users | Groups
EC2 Instances | EBS Volumes | Static IP's
EMR Clusters | Dynamo DB
NAT Gateways (optional)

When installing SYNTASA, it is recommended to tag all resources created so they can be tracked for billing purposes. This will also aid in the calculation of costs and the ability to expand or restrict resources as needed.

For any assistance with access controls or services required, contact your SYNTASA representative to assist with the necessary pre-requisites.

3. SYNTASA AWS ARCHITECTURE

AWS ARCHITECTURE OVERVIEW

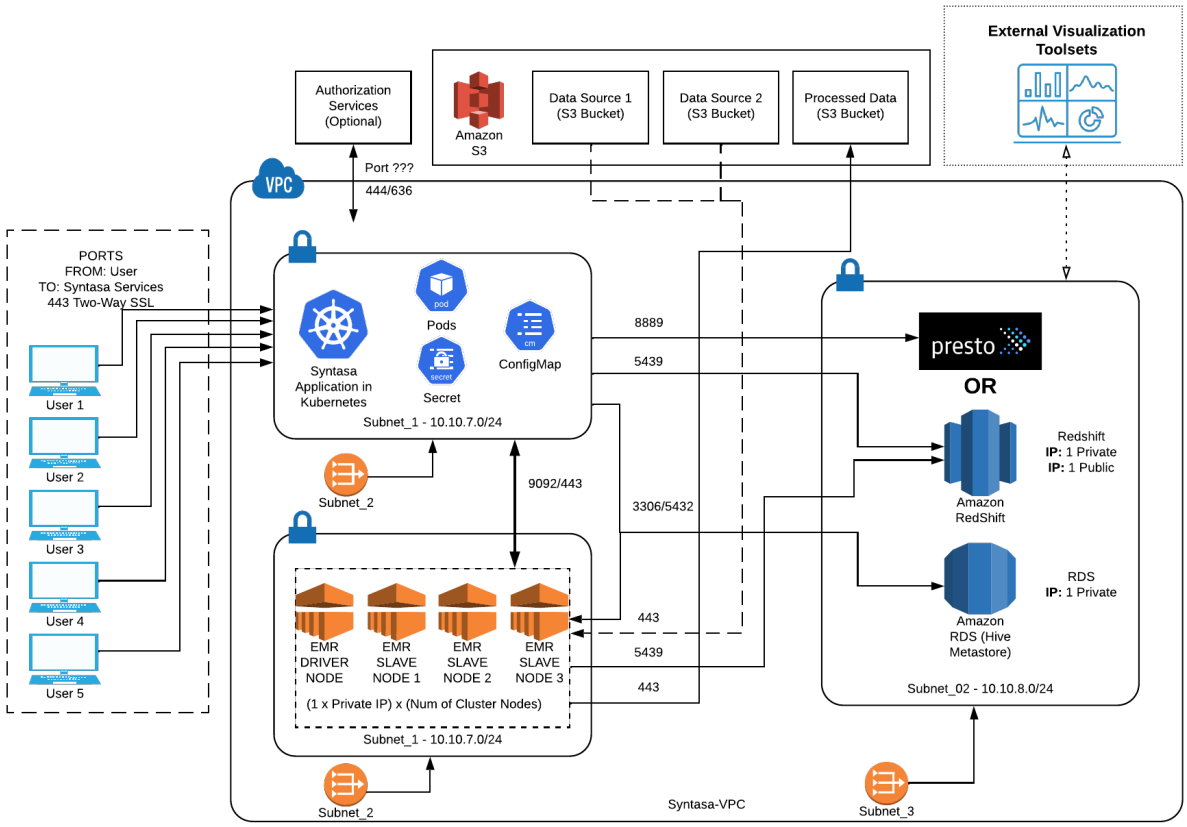


Figure 1 – AWS Infrastructure Overview Diagram. This diagram is just for reference purposes only, actual implementation may vary based on cloud resources/services available.

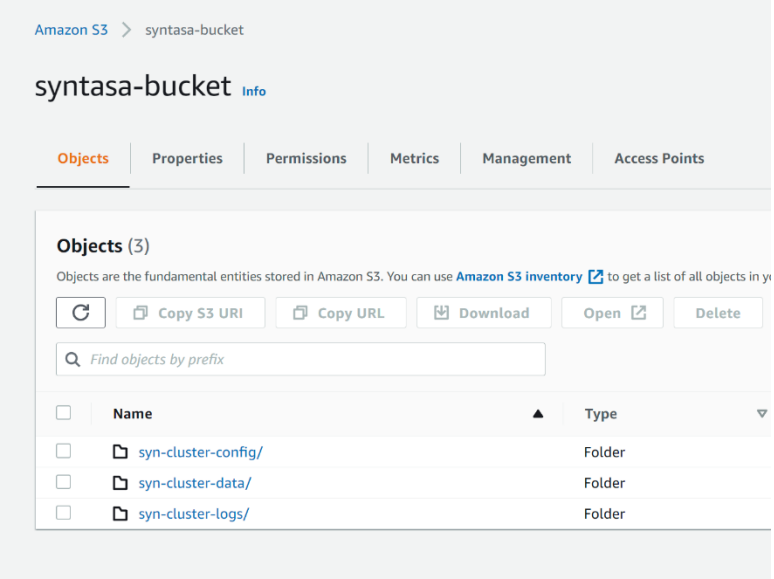
The SYNTASA architecture diagram above is a basic overview of the different services that can be used when setting up the platform in an AWS environment. There are optional items and some items that can be `either or` (like the output databases on the right side of the diagram). Use this for reference purposes to get a general idea of the cloud services that will be used as well as network ports and traffic.

Note – All names for resources can be set appropriately per your organizations' naming conventions and policies (unless explicitly noted that the names must be exactly as specified).

4. AMAZON S3 BUCKETS

The SYNTASA platform will need access to an S3 Bucket for storage of configuration files, log files, as well as processed data files. Note that the name used for the bucket can be set by the customer (especially if there are naming conventions and/or security requirements set by the organization). The “folders” or objects within the buckets will need to be as described below. **Note** - For the purposes of this document and installation, we will use a placeholder name for the bucket so that it can be referenced in the IAM Roles and Policies Sections in the sections that follow.

Bucket Name: syntasa-bucket
Folder #1: syn-cluster-config
Folder #2: syn-cluster-data
Folder #3: syn-cluster-logs



Additionally, set the permissions and encryption for the bucket as required by your organization.

5. AMAZON POLICIES AND IAM ROLES

S3 ACCESS POLICY

To access the bucket mentioned in Section 4, create a policy to access that bucket and its contents. Below is a sample policy created to give access to the above bucket. **Note** – this policy does have "list buckets" access so that the buckets can be listed, but the get/put/listacl etc… permissions are restricted to the "syntasa-bucket" object only.

Policy Name: syntasa_s3_access_policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:ListAllMyBuckets",
      "Resource": "*"
    },
    {
      "Sid": "VisualEditor1",
      "Effect": "Allow",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::syntasa-bucket",
        "arn:aws:s3:::syntasa-bucket/*"
      ]
    }
  ]
}
```

Figure 2 - Sample S3 Policy for access to S3 Buckets

KUBERNETES NODE ROLE

When using rancher, we assume an instance role for the "worker" type nodes are already set ahead of time. This same node instance role can be used with some additional permissions/policies added or a new role can be created from scratch with the permissions and policies that Rancher requires plus additional SYNTASA policies.

Role Name:	rancher_worker_aws_instance_role
Trust Relationships:	ec2.amazonaws.com
Policy/Permission #1:	Rancher "worker" node policy requirements. As per Rancher setup documentation.
Policy/Permission #2:	"syntasa_s3_access_policy" (for access to SYNTASA bucket from above)
Policy/Permission #3:	AmazonElasticMapReduceFullAccess (for ability to create/delete/modify EMR clusters)

EMR DEFAULT ROLE

Default policy for the Amazon EMR Service Role. Used for creating and managing EMR Clusters.

Role Name: EMR_DefaultRole
Trust Relationships: elasticmapreduce.amazonaws.com
Policy/Permission #1: AmazonElasticMapReduceRole

EMR EC2 DEFAULT ROLE

Default policy for the EC2 Instances spun up by Amazon EMR Clusters.

Role Name: EMR_EC2_DefaultRole
Trust Relationships: ec2.amazonaws.com
Policy/Permission #1: "syntasa_s3_access_policy" (for access to SYNTASA bucket)
Policy/Permission #2: AmazonElasticMapReduceRole
Policy/Permission #3: AmazonElasticMapReduceforEC2Role
Policy/Permission #4: AmazonElasticMapReduceFullAccess

EMR AUTOSCALING DEFAULT ROLE

Default policy for the EC2 autoscaling policies for AWS EMR Clusters

Role Name: EMR_AutoScaling_DefaultRole
Trust Relationships: *elasticmapreduce.amazonaws.com* and *application-autoscaling.amazonaws.com*
Policy/Permission #1: "syntasa_s3_access_policy" (for access to SYNTASA bucket)
Policy/Permission #2: AmazonElasticMapReduceRole
Policy/Permission #3: AmazonElasticMapReduceforAutoScalingRole

6. AMAZON VPC AND NETWORKS

VPC SETUP AND CONFIGURATION

The SYNTASA platform requires a VPC large enough to support 3 to 4 subnets (3 is the minimum, but 4 is preferred). Because we will be autoscaling clusters and starting multiple notebooks, one or two of the subnets will need to have enough IP address space to accommodate. Also, ensure subnets are in the same region. The exact requirements will be listed below in the subnets section.

RANCHER WORKER SUBNET 1

The SYNTASA application pods will be hosted on a worker node from within the Rancher RKE Cluster. This node(s) will be able to auto scale to accommodate increases in users and service calls.

Subnet Name: rancher_subnet_1
IP Addresses Required: *Requires at least 1 IP address with a max of 8 or 16. Minimum [/28] Preferred [/24]*

RANCHER WORKER SUBNET 2

The SYNTASA notebook and container pods will be hosted on a worker node from within the Rancher RKE Cluster. This node(s) will be able to auto scale to accommodate increases in users and notebook requirements as needed. If possible, make this subnet as big as possible (as resources allow) to accommodate multiple notebooks for users and ad-hoc container process jobs.

Subnet Name: rancher_subnet_2
IP Addresses Required: *Requires at least 128 IP addresses. Minimum [/24] Preferred [/23]*

RDS INSTANCES SUBNET

RDS is required for hosting Metastore instances (RDS Mysql and RDS Postgres). If preferred, this subnet can also be merged with the previous “Rancher Worker Subnet 1” to save on IP address space and for ease of setup.

Subnet Name: rds_subnet
IP Addresses Required: *Requires at least 2 IP address with a max of 8 or 16. Minimum [/29]*

EMR CLUSTER SUBNET

This is used for hosting EMR clusters that can have *N* number of nodes per cluster. These clusters can also auto scale to meet increases in data volume and computing resources for users.

Subnet Name: emr_cluster_subnet
IP Addresses Required: *Requires at least 128 IP addresses. Minimum [/24] Preferred [/22]*

SUBNET ROUTE TABLES

Make sure the subnets above have routes to be able to talk to each other and can talk to any external resources as necessary (such as python repositories, docker repositories, external authentication/authorization services, etc.)

7. AMAZON SECURITY GROUPS

For allowing communication between the different subnets, we are operating under the assumption that the subnets in the previous Amazon VPC and Networks section will be created or used for SYNTASA purposes explicitly. If this is not the case and port(s) need to be explicitly defined in the security groups below, this can be accommodated as well. Syntasa engineers can provide additional details as required.

RANCHER WORKER SECURITY GROUP

This will be applied to the nodes where the SYNTASA application pods will sit. This can be applied to multiple nodepool's as well.

Security Group Name:	rancher-worker-node-sg
SG Rule #1:	All worker group ports necessary for Rancher (e.g. etcd, ingress, Kubernetes api, etc.)
SG Rule #2:	All internal communication from the Rancher Worker Subnet 1
SG Rule #3:	All internal communication from the Rancher Worker Subnet 2
SG Rule #4:	All internal communication from the RDS Instances Subnet
SG Rule #5:	All internal communication from the EMR Cluster Subnet

RDS SECURITY GROUP

This will be applied to Metastore instances that will be created in the next step. This will allow communication to the RDS instances on the default Postgres and Mysql ports.

Security Group Name:	syntasa-rds-sg	
SG Rule #1:	**Description** MySQL Access to application pod nodes and EMR clusters	
	Protocol TCP	
	Port Range 3306	
	Source(s)	Rancher Worker Subnet 1 (MYSQL Access)
		Rancher Worker Subnet 2 (MYSQL Access)
		EMR Cluster Subnet (MYSQL Access)
SG Rule #2:	**Description** Postgres Access to application pod nodes and EMR clusters	
	Protocol TCP	
	Port Range 5432	
	Source(s)	Rancher Worker Subnet 1 (MYSQL Access)
		Rancher Worker Subnet 2 (MYSQL Access)
		EMR Cluster Subnet (MYSQL Access)

EMR MASTER NODE SECURITY GROUP

Allows internal access between the different nodes of an EMR cluster.

Security Group Name:	syntasa-emr-master-sg
SG Rule #1:	All internal communication from the Rancher Worker Subnet 1
SG Rule #2:	All internal communication from the Rancher Worker Subnet 2
SG Rule #3:	All internal communication from the RDS Instances Subnet
SG Rule #4:	All internal communication from the EMR Cluster Subnet

EMR WORKER NODE SECURITY GROUP

Allows internal access between the different nodes of an EMR cluster.

Security Group Name:	syntasa-emr-worker-sg
SG Rule #1:	All internal communication from the Rancher Worker Subnet 1
SG Rule #2:	All internal communication from the Rancher Worker Subnet 2
SG Rule #3:	All internal communication from the RDS Instances Subnet
SG Rule #4:	All internal communication from the EMR Cluster Subnet

EMR SERVICE NODE SECURITY GROUP

Allows internal access between the different nodes of an EMR cluster. This is used when running clusters in a private subnet (without external network communication).

Security Group Name:	syntasa-emr-service-sg
SG Rule #1:	All internal communication from the Rancher Worker Subnet 1
SG Rule #2:	All internal communication from the Rancher Worker Subnet 2
SG Rule #3:	All internal communication from the RDS Instances Subnet
SG Rule #4:	All internal communication from the EMR Cluster Subnet

NOTE: These security groups for EMR are placeholders that will be passed to the Amazon EMR API when creating clusters. Upon the startup of the first cluster, EMR will auto-populate rules into these groups as it requires.

LOAD BALANCER SECURITY GROUP

When using a load balancer please make sure that external access to port 443 is allow. Additionally, below external access is listed as the entire world, e.g., 0.0.0.0/0. It is recommended that this be whittled down to the external CIDR ranges for client access.

Security Group Name:	load-balancer-sg
SG Rule #1:	**Description** Access to the SYNTASA application on port 443
	Protocol TCP
	Port Range 443
	Source(s) 0.0.0.0/0

8. AMAZON RDS METASTORES

The SYNTASA Setup requires two RDS metastores to be present in the AWS Account, one Postgres instance which the Application will use to store its internal metadata (access information, user accounts, job executions etc.) and a second metastore, MySQL, that will be used for Apache Hive Data Catalogs.

SYNTASA APPLICATION METASTORE

This RDS Instance will be used to store internal SYNTASA application metadata.

RDS Instance Identifier:	syntasa-application-metastore
RDS Type:	PostgreSQL
RDS Version:	Postgres 9.6+ (suggested to use 11.15-R1 and above)
RDS Instance Class:	*db.t3.medium* or *db.m3.medium*
RDS Storage:	50 GB auto scaled to 200 GB max
Template:	Dev/Test
Master Username:	postgres
Master Password:	*<auto_generate>* or follow standard organizational password policy requirements
VPC:	VPC from the Amazon VPC and Network section
Subnet:	RDS Instances Subnet from the Amazon VPC and Network section
Security Group:	RDS Security Group from the Amazon Security Groups section
Initial Database Name:	syntasa
DB Parameter Group:	default (or if required, a new one can be created)
Backups :	enabled (7-day retention period)
Encryption:	enabled (default KMS key is preferred)

SYNTASA APPLICATION METASTORE

This RDS Instance will be used to store SYNTASA application metadata.

RDS Instance Identifier:	syntasa-hive-metastore
RDS Type:	MySQL
RDS Version:	MySQL 5.7+ (suggested to use 8.0 and above)
RDS Instance Class:	*db.t3.medium* or *db.m3.medium*
RDS Storage:	50 GB auto scaled to 200 GB max
Template:	Dev/Test

Master Username:	admin
Master Password:	*<auto_generate>* or follow standard organizational password policy requirements
VPC:	VPC from the Amazon VPC and Network section
Subnet:	RDS Instances Subnet from the Amazon VPC and Network section
Security Group:	RDS Security Group from the Amazon Security Groups section
Initial Database Name:	hive-metastore
DB Parameter Group:	default (or if required, a new one can be created)
Backups :	enabled (7-day retention period)
Encryption:	enabled (default KMS key is preferred)

9. ADDITIONAL SETUP ITEMS

EC2 Key Pair

To spin up EMR clusters an EC2 key pair is required, create an EC2 key pair for this instance. If an existing key pair is being used for access to instances for the project and you would like to keep that the same, that can be used as well.

Installation Server Instance

When installing and setting up Syntasa, it is recommended that an "installation server" be hosted that can be accessed via ssh and can access all the above resources including any Kubernetes and Docker endpoints. This is also where the installation packages can be kept, and installation referenced from. After installation, this instance can be shut down to save on costs.

Recommended Instance Type:	t3.medium (or t3.small).
Instance EBS Volume:	100 to 200GB (Since the Syntasa installation packages can be large, ensure ample store is allocated).

Kubernetes RKE Cluster

To install the components of SYNTASA please create an RKE cluster configured to the specifications of your organization. If a private registry is being used for Docker, please configure that in the Cluster settings as well. The SYNTASA engineers will need access to the RKE Cluster after it is created.

Kubernetes RKE Node Pools and Sizes

For initial node sizes (if cluster autoscaling is not available), ensure the following Node Templates or Node Pools are available.

Syntasa Application Node Group

Recommended Instance Type:	r5.2xlarge or r5a.2xlarge (8 cores and 64GB)
Instance EBS Volume:	100 to 200GB
IAM Instance Profile Role:	Kubernetes Node Role from Amazon Policies and IAM Roles Section.
Machine AMI:	Any preferred Debian/Redhat Flavors.
Node Selector Labels:	*Key:* syn-system-non-scalable *Value:* owned
	Key: syn-system-scalable *Value:* owned

Syntasa Container and Notebook Node Group

Recommended Instance Type:	t3.medium, t3.large, t3.xlarge, m5.large, m5.xlarge
Autoscaling Policy:	Karpenter should spin up and assign pods to any pod that has the below Node Selector Label
Instance EBS Volume:	25GB to 50GB
IAM Instance Profile Role:	Kubernetes Node Role from Amazon Policies and IAM Roles Section.
Machine AMI:	Any preferred Debian/Redhat Flavors.
Node Selector Labels:	*Key:* syn-name *Value:* syn-jupyter-nodegroup

Kubernetes RKE Node Pools and Sizes

If using Rancher RKE to automatically spin up EBS Volumes for Persistent Volumes, create the following Storage Classes/Persistent Volumes/Persistent Volume Claims in the Rancher UI. Optionally, this can be done by SYNTASA Engineers if granted access to the Rancher UI.

Storage Classes

Name:	syntasa-standard
Volume Type:	GP2 (general purpose SSD)
File System:	ext3 or ext4
Availability Zone:	automatic
Encryption:	disabled (can be enabled if need be).
Reclaim Policy:	retain the volume for manual clean up

Volume Expansion: enabled
Volume Binding Mode: Bind once PV Claim is Created.

Persistent Volume 1

Name: can specify one or have this created automatically by Rancher using the storage class above
Access Mode: Single Node Read-Write
Storage Class: syntasa-standard storage class from previous section
Volume Plugin: Amazon EBS Disk
Capacity: 50GB
File System: ext4

Persistent Volume 2

Name: can specify one or have this created automatically by Rancher using the storage class above
Access Mode: Single Node Read-Write
Storage Class: syntasa-standard storage class from previous section
Volume Plugin: Amazon EBS Disk
Capacity: 50GB
File System: ext4

Persistent Volume 3

Name: can specify one or have this created automatically by Rancher using the storage class above
Access Mode: Single Node Read-Write
Storage Class: syntasa-standard storage class from previous section
Volume Plugin: Amazon EBS Disk
Capacity: 50GB
File System: ext4

Persistent Volume Claim 1

Name: syntasa-frontend-pv-claim
Namespace: syntasa
Access Mode: Single Node Read-Write
Persistent Volume: Bind to Persistent Volume 1

Persistent Volume Claim 2

Name: syntasa-backend-pv-claim
Namespace: syntasa
Access Mode: Single Node Read-Write
Persistent Volume: Bind to Persistent Volume 2

Persistent Volume Claim 3

Name: syntasa-platform-pv-claim
Namespace: syntasa
Access Mode: Single Node Read-Write
Persistent Volume: Bind to Persistent Volume 3

NOTE: If namespace "syntasa" doesn't exist yet, create it using the kubectl command line tool from the installation server specified in Additional Setup Items.

Command: "*kubectl create namespace syntasa*"

INFRASTRUCTURE SETUP COMPLETE

At this point, the infrastructure setup for the SYNTASA components is complete. The next steps are to install the SYNTASA application into the RKE Cluster. Refer to the SYNTASA installation document for next steps. For any questions or support, please contact:

Kaushik Vinjamuri – kaushik.vinjamuri@syntasa.com
Michael Zaun – michael.zaun@syntasa.com
Brian Pavlicek – brian.pavlicek@syntasa.com

OUR OFFICES

HEADQUARTERS
560 Herndon Parkway, Suite 240
Herndon, VA 20170

LONDON
3 Lloyd's Avenue, 3rd Floor
London, EC3N 3DS
United Kingdom

syntasa.com | info@syntasa.com | @SYNTASACO
linkedin.com/company/syntasa | facebook.com/syntasa

SYNTASA®