Lab 13. Deploy Tableau Server

In this lab, we will cover the following recipes:

- Deploying Tableau Server in Windows
- Deploying to Tableau Server
- Deploying Tableau Server on Linux using AWS
- Getting started with Tabcmd

Technical requirements

There are multiple ways available to get the Tableau Server, namely:

- Deploy on-premise
- Deploy on cloud using AWS, Azure, or Google Cloud Platform
- Get a preconfigured Tableau Server at the Cloud Marketplace
- Use Tableau Online In our case, we will be using AWS, and we will deploy Tableau Server on top of an EC2 instance, to get the advantage and flexibility of the cloud, and to demonstrate the main concepts of Tableau Server.

Introduction

Tableau Server is a core element of any analytical solution. It helps to share insights across organizations and allows end users to access enterprise data. It is secure and scalable. Moreover, the recently released Tableau Server is available on Windows and Linux.

It is good to know how Tableau licesing their Server. There are two options available:

- User-based - you need license for every user who access Tableau Server. In addition, user based licnes has different options such as Viewer, Exploer and Creater.
- Core-based - you license total number of cores for your entire Tableau Server implementation. When you start to plan your Tableau Server implementation, you shoud start from planning hardware and licesnes based on business needs, such as:
- How many users does your organization have?
- How fast does your organization grow?
- What is the level of Tableau knowledge of users?
- How big is your Data Warehouse?
- and so on. You might learn more about license types here https://onlinehelp.tableau.com/current/server/en-us/license_server_overview.htm. It is applicable for Linux and Windows implementations. In addition, you may leran about the prices for license: https://www.tableau.com/pricing/teams-orgs.

This lab will guide you through the Tableau Server deployment on both Linux and Windows to give you an idea of the differences. In addition, it will cover key concepts of Tableau Server and best practices of deployment using AWS.

Deploying Tableau Server in Windows

In this recipe, we will learn how to self-deploy Tableau Server on AWS using Windows. Self-deploy offers the most flexibility in security, scaling, and capacity. Also, the total cost of ownership and the amount of time to deploy in the cloud versus on-premise solutions should be much less.

Getting ready

To complete this recipe, please have an AWS account and the Tableau Server product key.

For a single production instance, please be prepared to build an EC2 instance using the recommended resource requirements:

- Windows Server 2012 R2, 64-bit or Windows Server 2016, 64-bit
- 16+ vCPU
- 64+ GB RAM (4 GB RAM per vCPU)
- 30-50 GB for the the OS
- 100+ GB for Tableau Server
- EBS Storage type (SSD (gp2) or provisioned IOPS)
- Less than or equal to 20 milliseconds

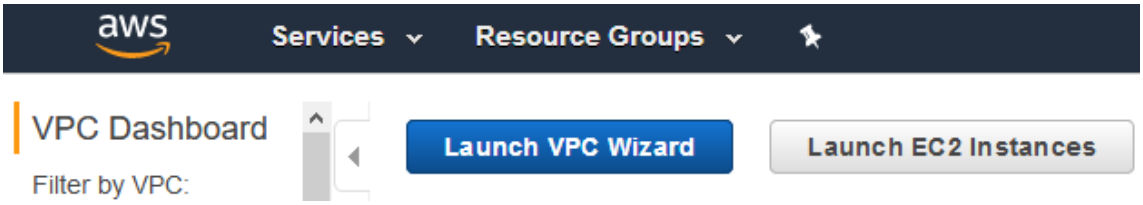
Note

The minimum hardware requirements are 64-bit processor, 16 vCPU, 32 GB RAM, 50 GB disk space. If the server does not have the minimum hardware requirements you will not be able to install Tableau Server.

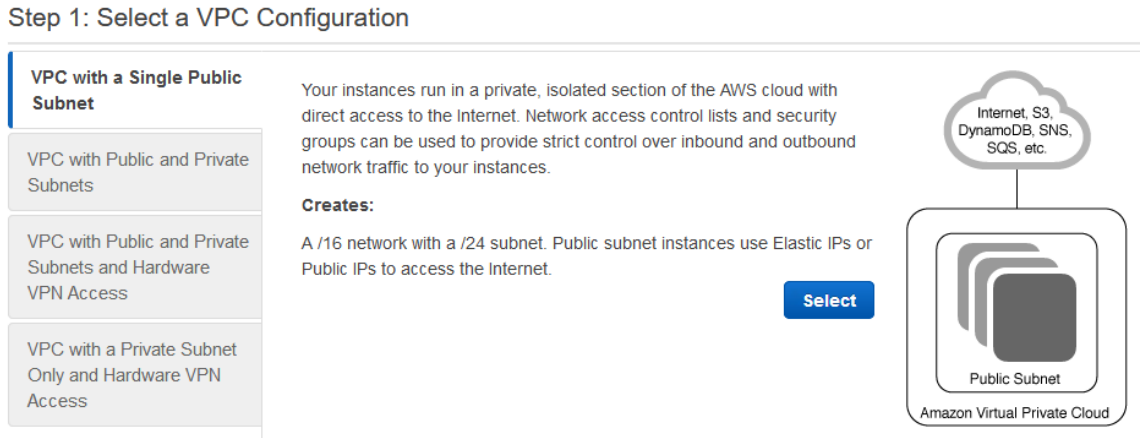
How to do it..

To begin the recipe, we need to log into AWS and take the following steps:

1. Create a [**Virtual Private Cloud**] ([**VPC**]).
2. Navigate to the Amazon VPC console.
3. Click on `Launch VPC Wizard`, as follows:



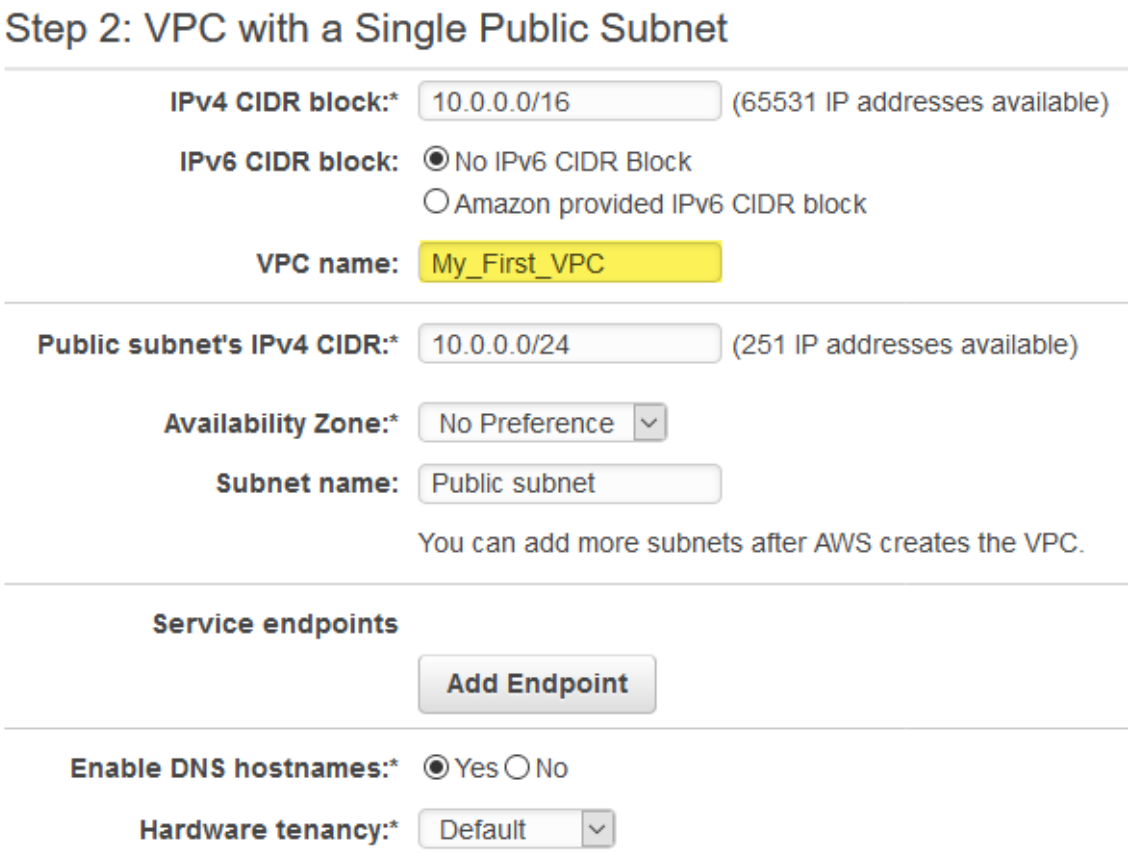
4. Select `VPC with a Single Public Subnet`, as follows:



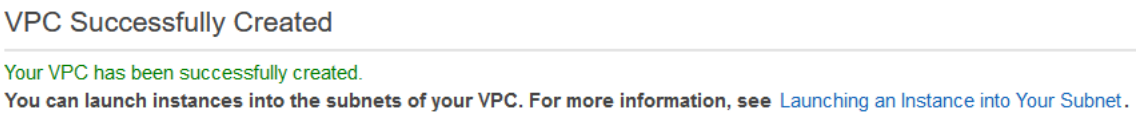
Note

Note: When we are creating VPC we should make sure, that we have Public Subnet with Internet Gateway and Router table. Otherwsise, you will be able access Tableau Server GUI only from VPC. You can read more about this at AWS documentation: https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html. In this lab, we didn't focus on creating Internet access for the server and used RDP and Private DNS to access Tableau Server.

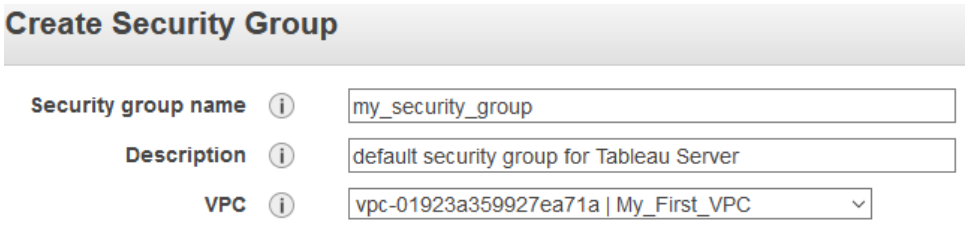
5. Give the VPC a name and click `Create VPC`, as follows:



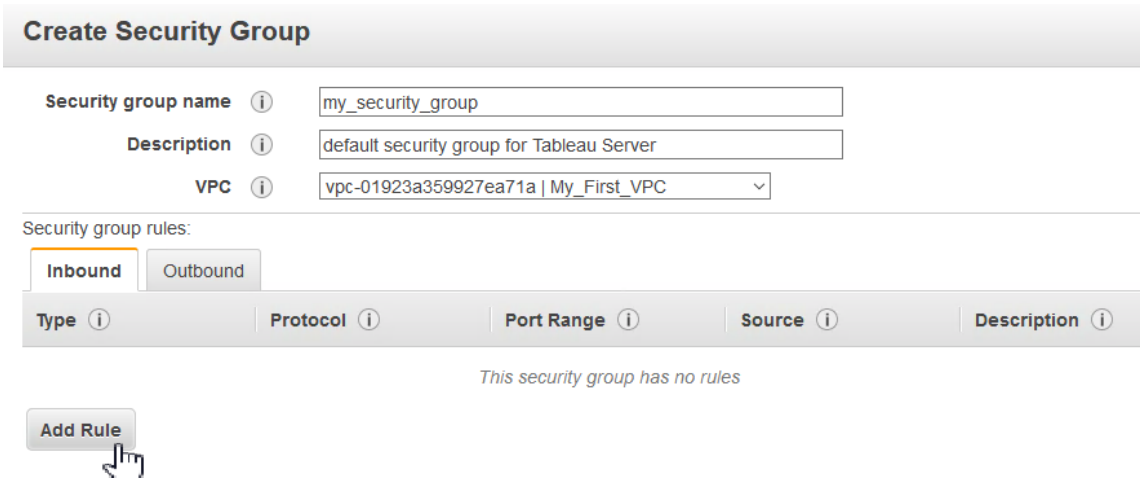
6. We will see our VPC created, as follows:



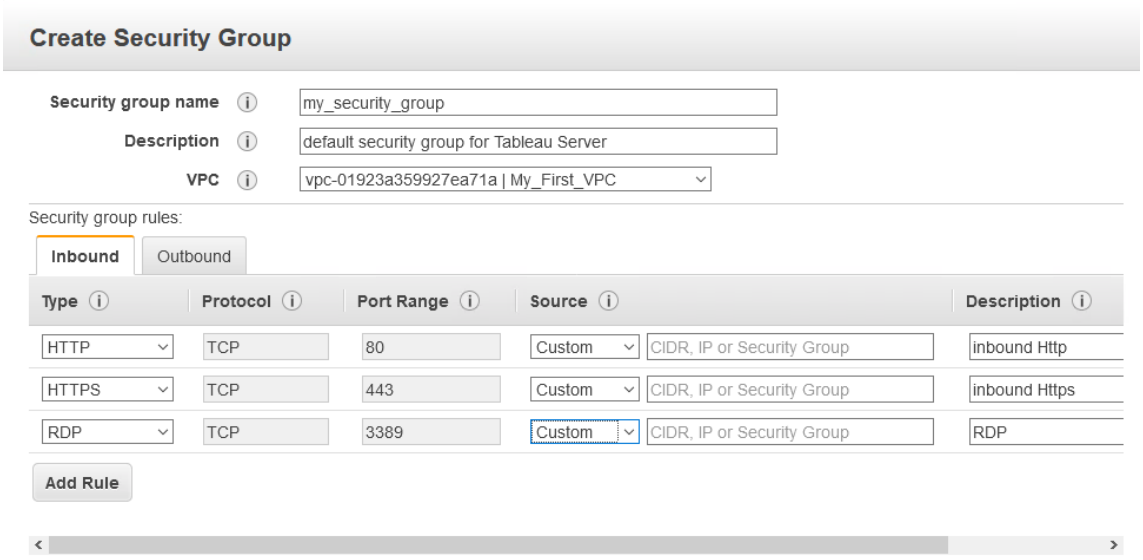
7. The next step will be to configure the networking and security. To configure networking and security, we will use the following steps:
    - Go to the EC2 console.
    - Ensure your region is in the same location as the VPC we just created.
    - Click `Security Groups` in the `Navigation` pane.
    - Click on `Create Security Group`.
    - Fill in the security group name, description, and select the VPC we created in the previous steps, as follows:



8. Create the inbound rules, by clicking on `Add Rule`, as follows:

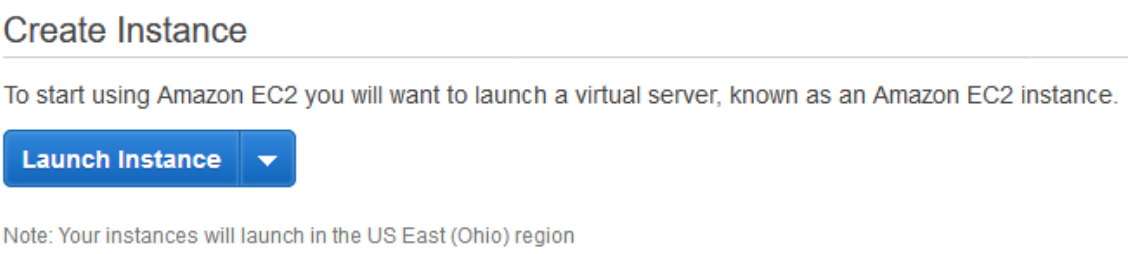


9. Create the inbound traffic rules for `HTTP`, `HTTPS`, and `RDP`. Then, click on `Create`, as follows:



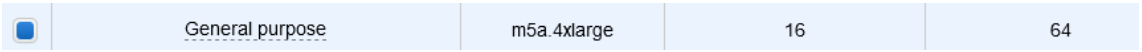
10. Use `Custom` to specify an IP range or another security group. For production environments, it is important to limit who has access to your environment. In addition, we should add SSH port `22` for the Linux environment. Finally, we need to add port `8850` for the Tableau Service Manager (TSM) web interface.
11. Launch an Amazon EC2 instance.
12. Navigate to the EC2 console.

13. Verify you are in the same region as the VPC created previously.
14. Under **`Create Instance`**, Click on **`Launch Instance`**, as follows:



15. Select an Amazon Machine image for Windows Server 2012 R2, 64-bit or Windows Server 2016, 64-bit, as follows:

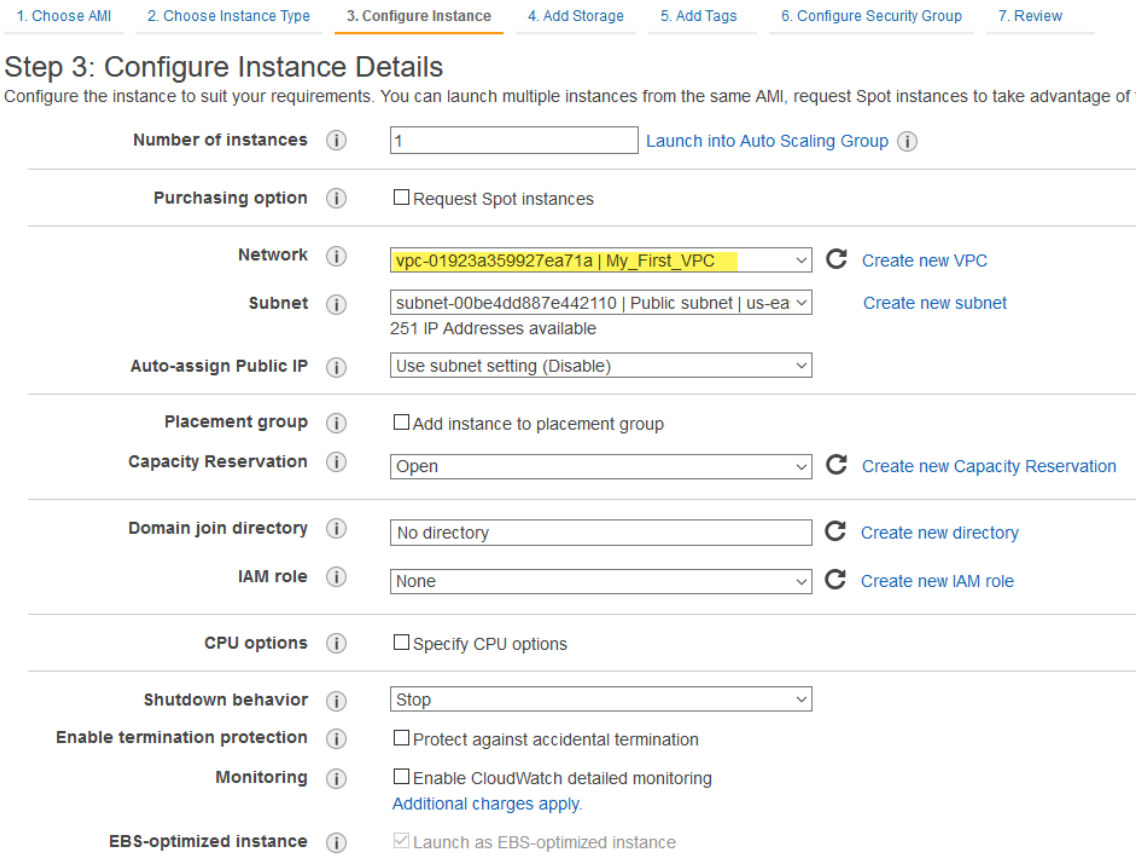
16. Withinthe **`Choose an Instance Type`** menu, scroll down to select an instance that meets the recommended requirement for Tableau Server: 16+ vCPU and 64+ GB RAM (4 GB RAM per vCPU), as follows:



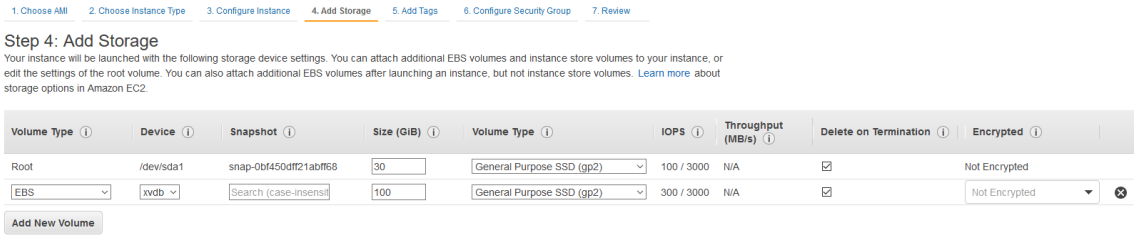
Note

Installing Tableau on Amazon EC2 T2 instances are not supported. Typical environment types and sizes for development, test, and production are as follows: `c5.4xlarge`, `m5.4xlarge`, `r5.4xlarge`

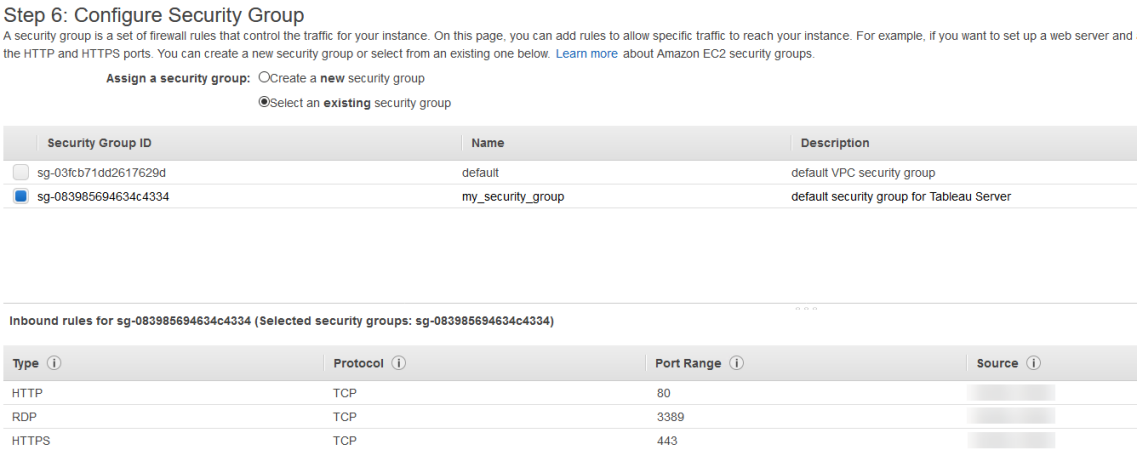
17. Configure the instance type by choosing the VPC created in the previous steps, as follows:



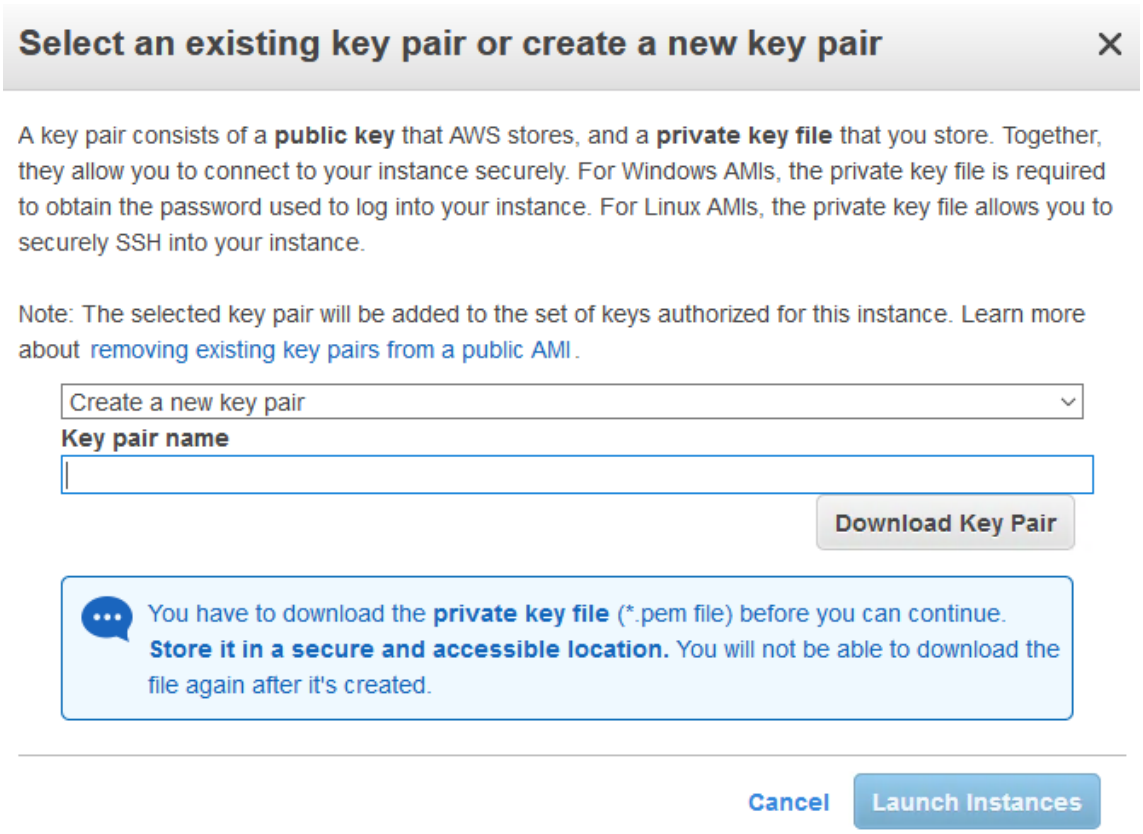
18. Add another 100 GB of storage as a separate drive. You may have to partition and mount this drive, as follows:



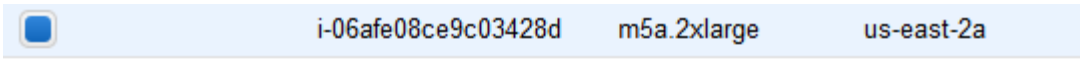
19. In the `Configure Security Group` menu, choose the security group we created in the previous recipe steps, as follows:



20. Review the instance you created and then launch it.
21. Create a key pair file (or use an existing one). We need to create this `.pem` file in order to log in to the server remotely, as follows:

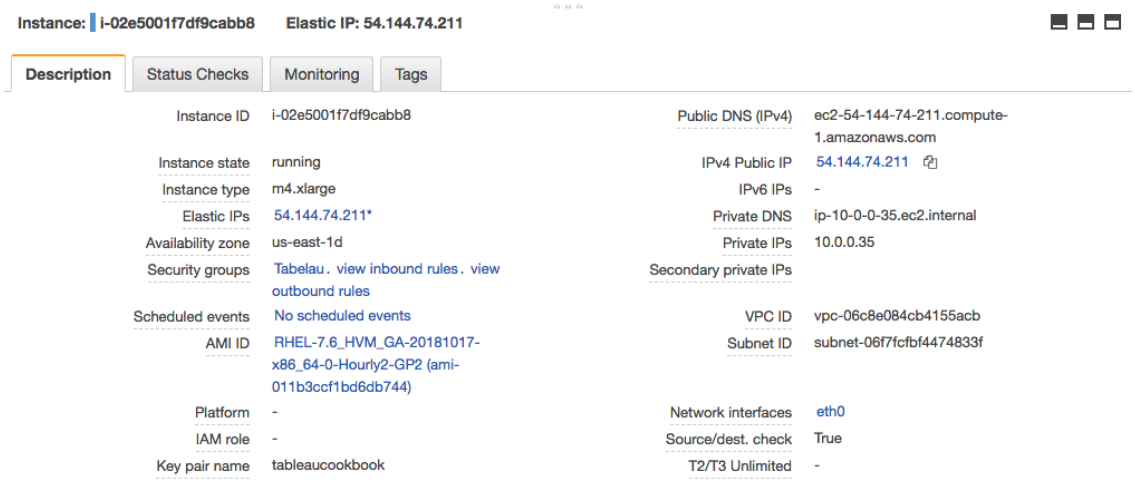


22. It can take a few minutes to launch the instance. While it is initializing, copy the instance ID for the next step, as follows:

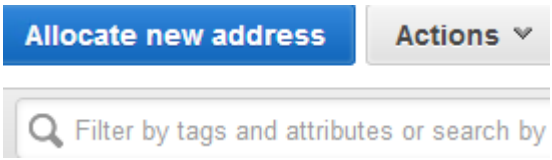


23. Create an elastic IP address for the VPC:

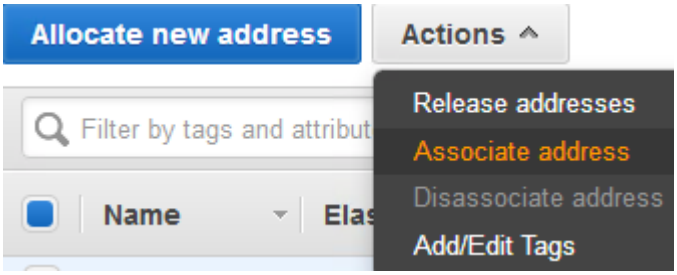
- Navigate to the Amazon VPC console
- Use the same region as the VPC created in earlier steps
- In the **`Navigation`** pane, choose elastic IP, as follows:



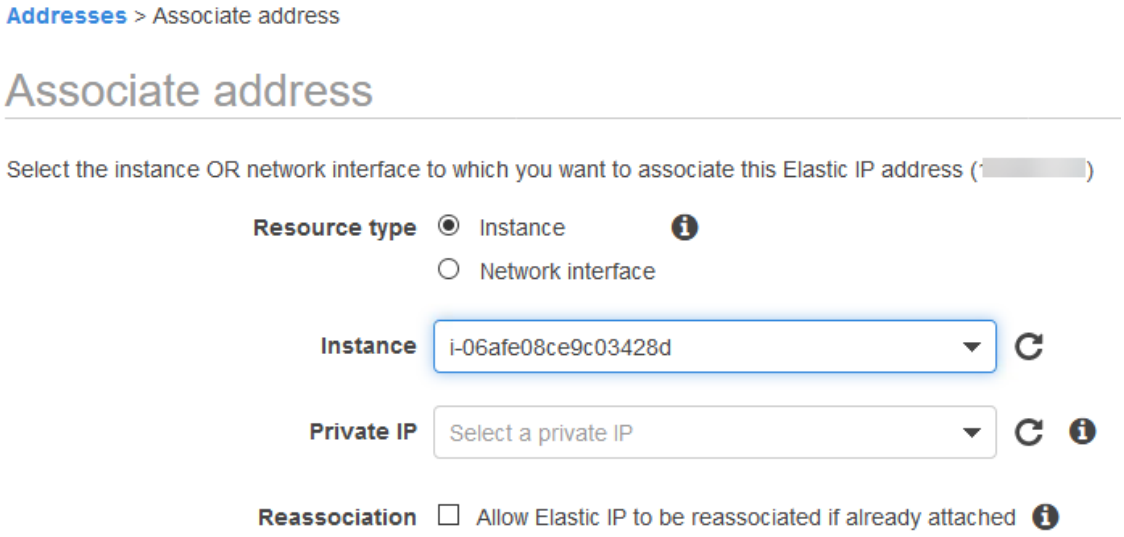
24. Click on the **`Allocate new address`** button and then click on **`Allocate`** in the next screen, as follows:



25. Once created, click on the **`Actions`** menu, and select **`Associate address`** , as follows:

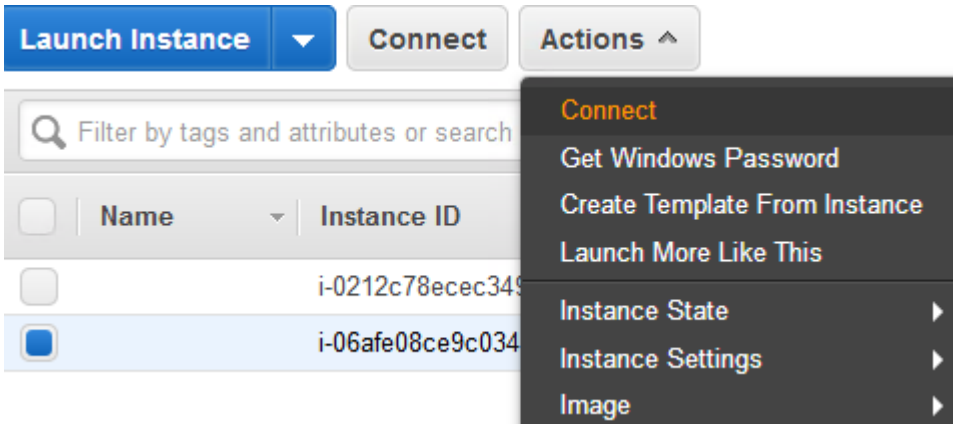


26. In the **`Associate address`** window, choose the **`Instance`** resource type and the server/instance you created in the earlier steps, as follows:



27. Log into Amazon EC2 using remote desktop:

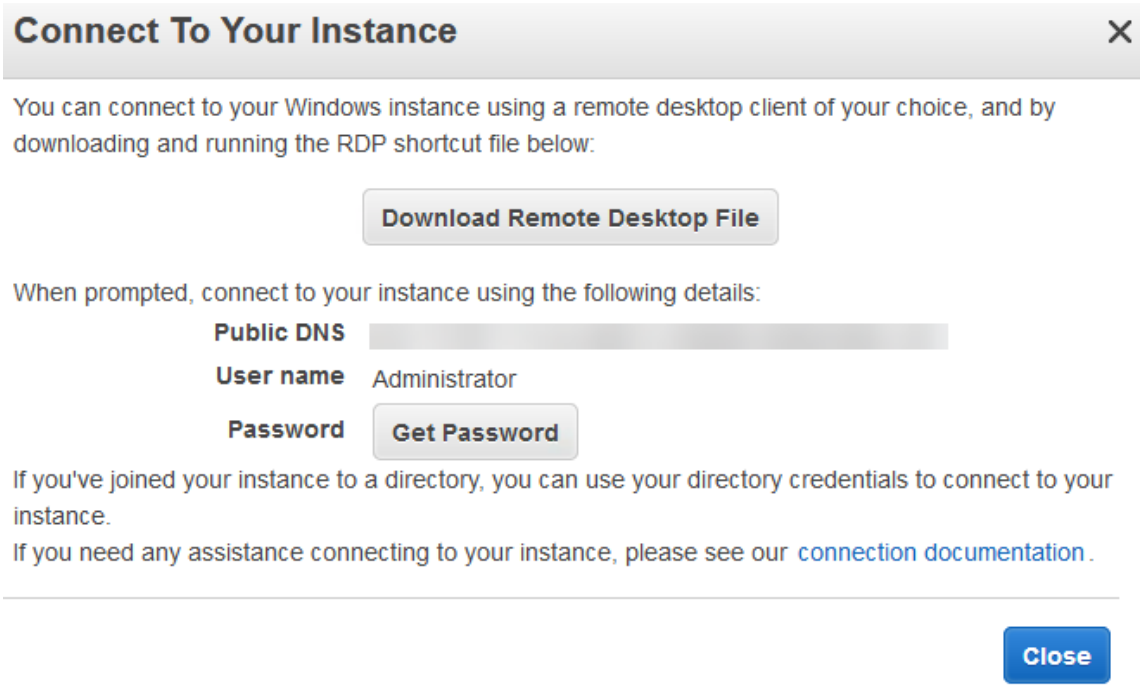
    - Navigate to the EC2 console.
    - Select the region where we launched the instance.
    - In the EC2 dashboard, click on Instances.
    - Select the instance, click on **`Actions`**, and choose **`Connect`**, as follows:



28. In the **`Connect To Your Instance`** dialog box, click on **`Download Remote Desktop File`**.

29. Click on **`Get Password`** and select the `.pem` file created earlier.

30. Click on **`Decrypt Password`**, when it's displayed copy it and keep it, as follows:

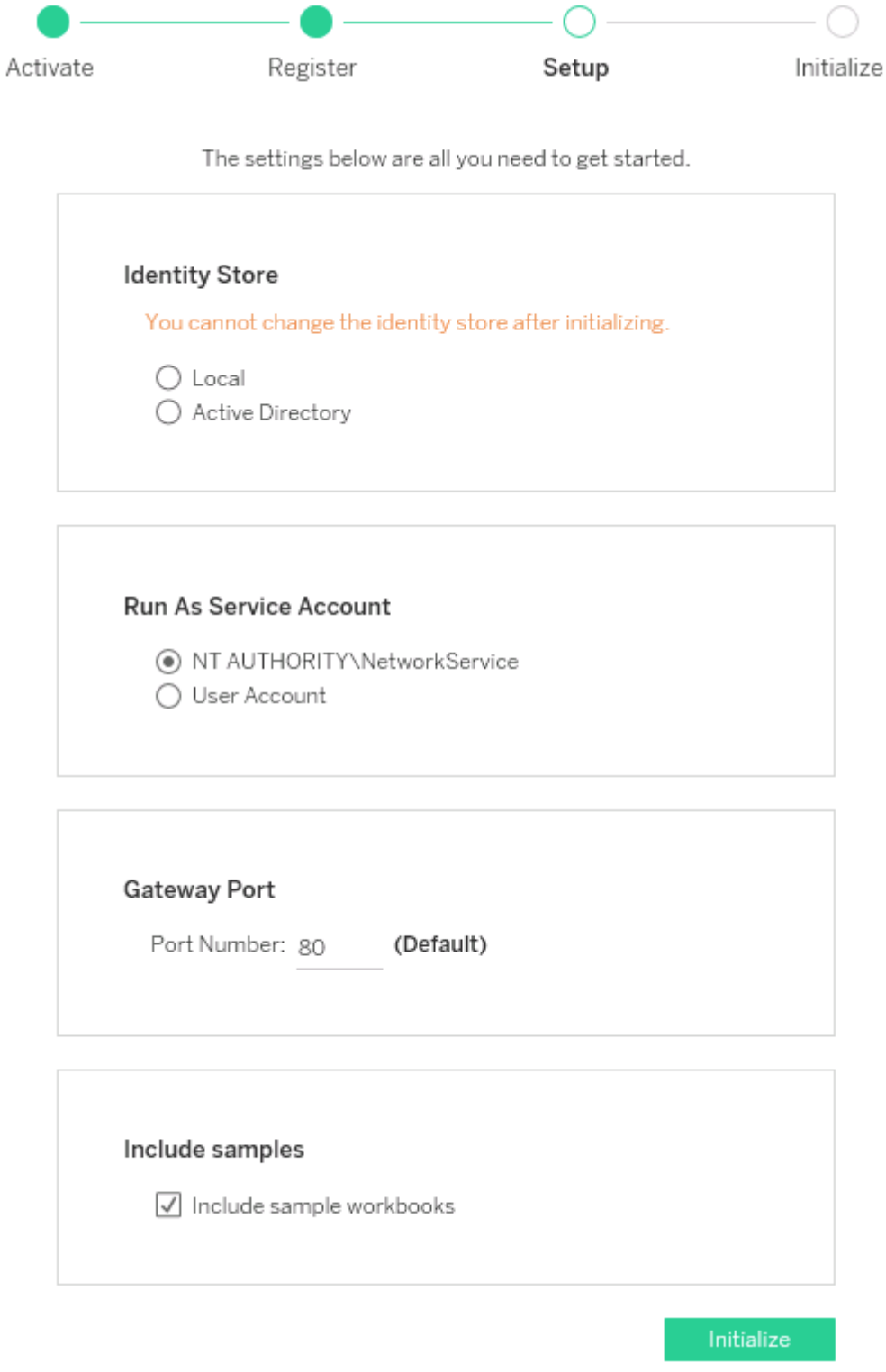


31. Log in using the `.rdp` file saved earlier and ignore any messages or warnings about unknown publishers or unknown connections.

32. Install Tableau Server as follows:

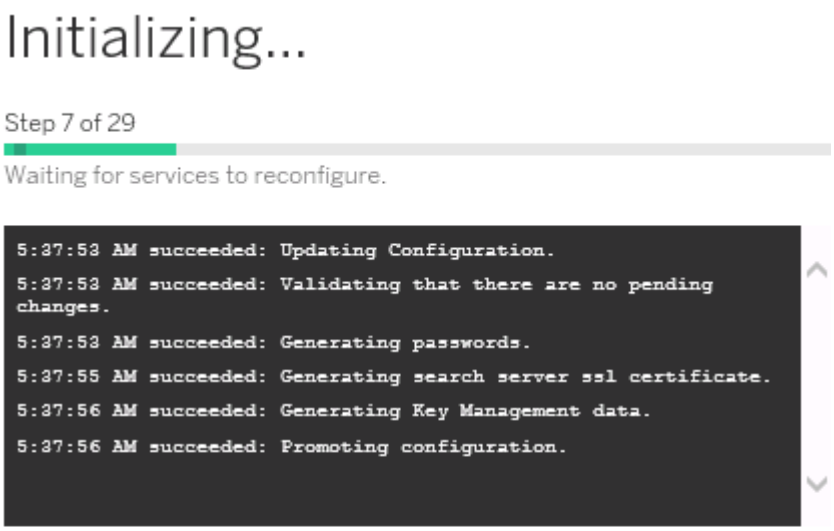
    - Download the installer for Tableau Server 2019.x and save it locally
    - For a single node, guided install, double-click on the installer and follow the instructions to create a new Tableau Server installation. Rather than installing onto the `C` drive, it is best practice to install the application on a separate drive. We added this in the previous steps for setting up an EC2 instance

    33. Once Tableau Server has been installed, you must use an account with local administrator rights to run [**Tableau Services Manager**] ([**TSM**]) Web UI, and CLI tools. Follow the guided instructions to activate and register Tableau Server. Go through the following setups in the following screenshot:

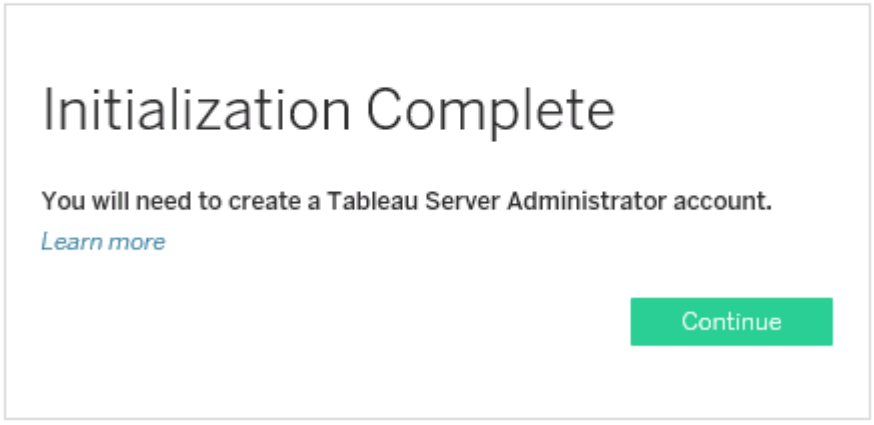


34. Click

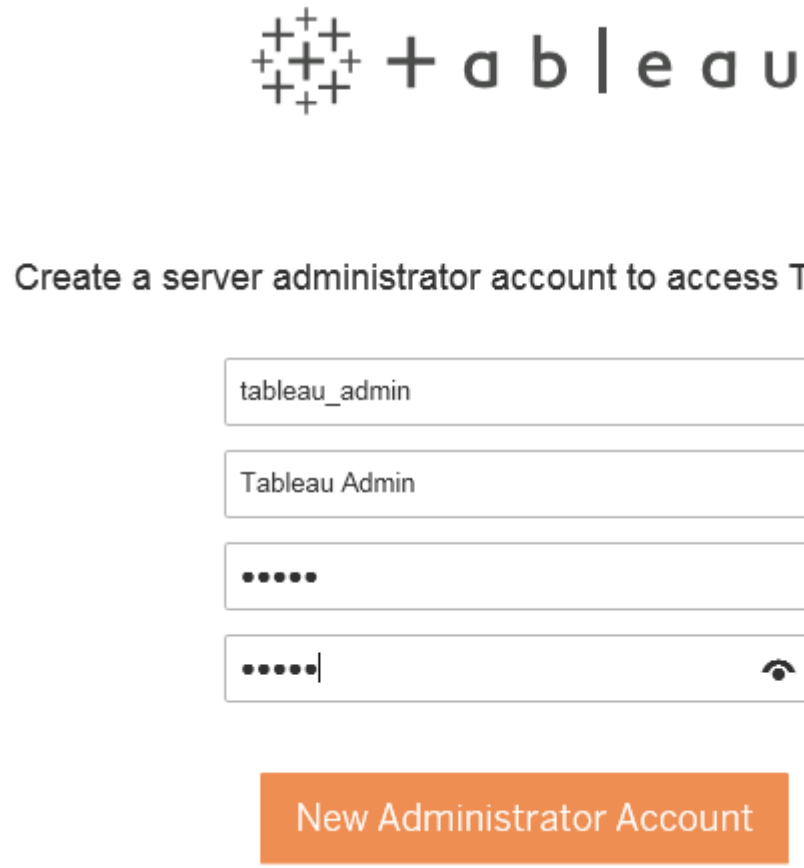
35. on `Initialize`:



36. Click on `Continue`:



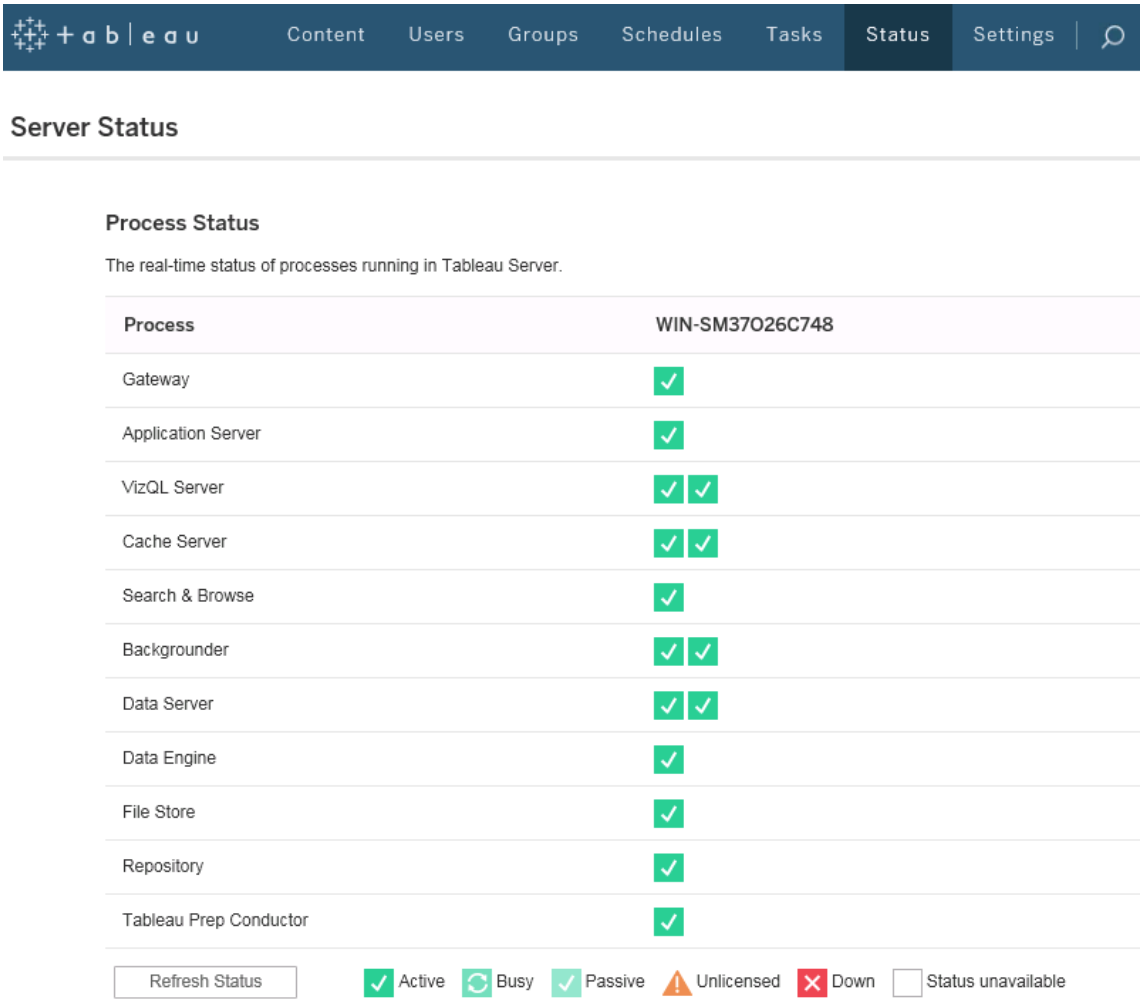
37. Create a tableau administrator account, as follows:



Note

Note: By default, Tableau will using Local Authentification method. However for the Production purpose it is good to use another authentification methods such as AD, Kerberos and so on. You might learn more about it using Tableau documentation: https://onlinehelp.tableau.com/current/server/en-us/security_auth.htm. You may use TSM CLI or TSM GUI in order to setup authentification method and you should do it during initializing of Tableau Server. You can't switch later to another method.

38. You can validate the install by going to the built-in administrative views in Tableau, as follows:



Note

Note: This is default Tableau Server configuration for Single node. You may increase number of backgrounders, for example if you have enought capasity. This is advance topic and it depends on organization needs. You may learn more about sizing of Tableau Server in White Paper Tableau Server Scalability - A Technical Deployment Guide for Server Administrators Read more at https://www.tableau.com/learn/whitepapers/tableau-server-scalability-technical-deployment-guide-server-administrators#OcvXkMS7csDuTIRu.99{.ulink}

How it works...

With AWS we created the infrastructure and server we needed for Tableau Server. We created a virtual private cloud with a private subnet. We created a security group within our VPC, so we can accept web requests and remote desktop into the server. We configured an EC2 instance and launched it. Next, we downloaded the Tableau Server installer and executed it. We used the new TSM to configure Tableau Server.

There's more...

We can use an automated installer `SilentInstaller.py`. It's written in Python and community supported. Automated installers are great to use when you need multiple servers or environments to have the same configuration. See Tableau's online help for more instructions at https://onlinehelp.tableau.com/current/server/en-us/automated_install_windows.htm.

In addition, we want to mention TSM. It is a new feature that was introduced with version 2018 R2 on Windows and with version 10.5 on Linux. It serves the configuration and administration purpose of Tableau Server. You can learn more about TSM here:

- [**Windows**]: https://onlinehelp.tableau.com/current/server/en-us/tsm_overview.htm.
- [**Linux**]: https://onlinehelp.tableau.com/current/server-linux/en-us/tsm_overview.htm. The downside of this tool is, that if you have Tableau Server without TSM (older version) and want to upgrade, you have to install Tableau Server and install a fresh copy of new version with TSM, as explained here:

https://onlinehelp.tableau.com/current/server/en-us/sug_pretsm_to_tsm.htm.

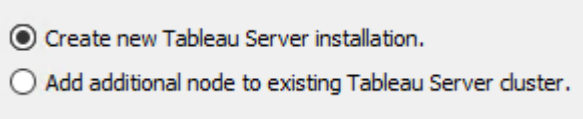
Read more about coordination services and configuring processes in the second and third nodes in the online Tableau help document at https://onlinehelp.tableau.com/current/server/en-us/distrib_ha_install_3node.htm.

See also

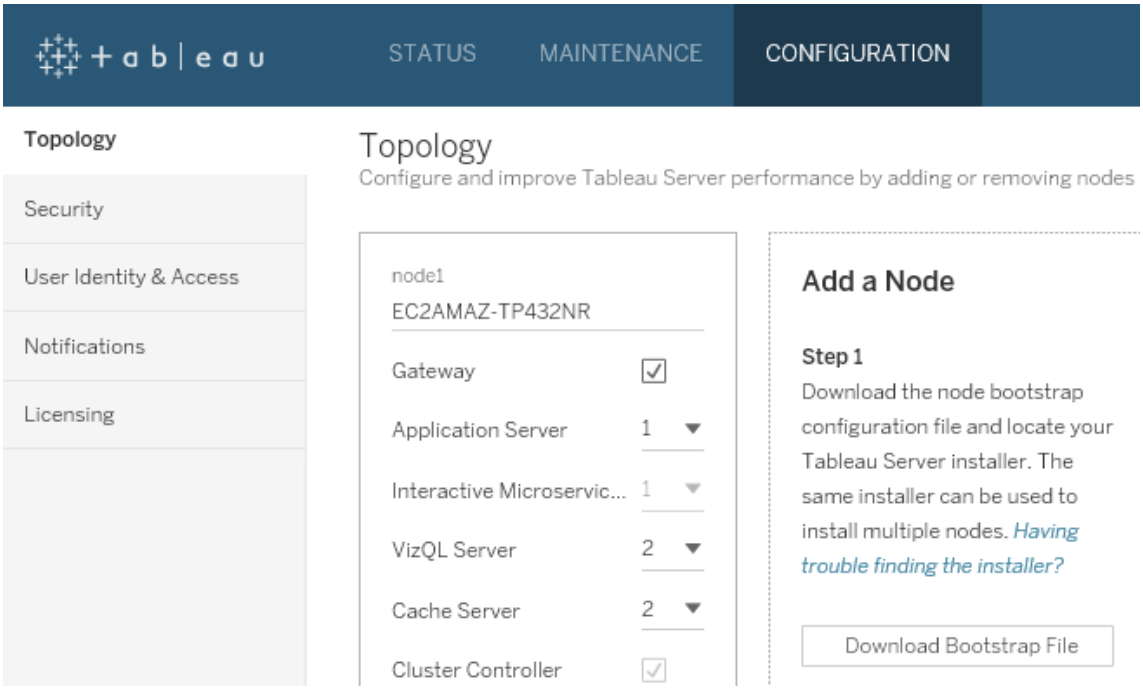
There are other methods that we can use to install Tableau Server. One way is to build a clustered environment. This architecture is great for systems that have 24 hours service level requirements or to optimize performance.

This recipe assumes you've stood up the correct architecture components in AWS and followed the best practice guidelines for network security and load-balancing:

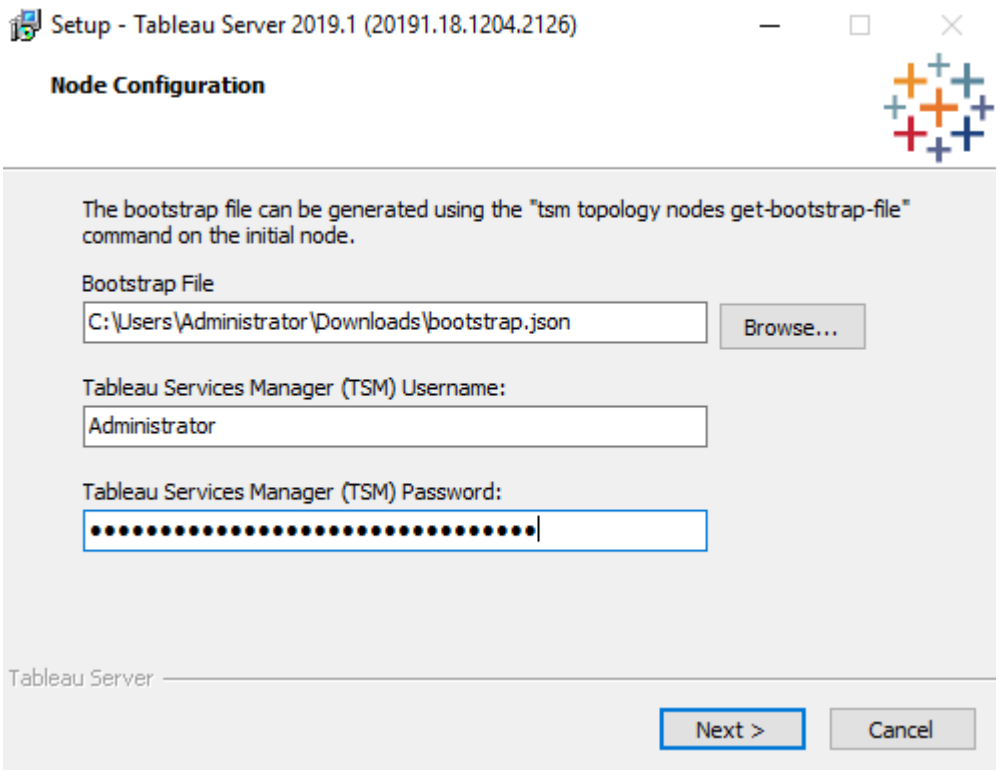
1. When installing Tableau on the first node (referred to as instance 1), use the `Create new Tableau Server installation` option. When installing Tableau Server on the redundant nodes (referred to as instance 2), click on the additional node to existing Tableau Server cluster option, as follows:



2. Create a bootstrap file from instance 1, by logging into the TSM.
3. Go to `Configuration` and click on `Download Bootstrap File`, as follows:



4. Copy the `bootstrap.json` file to instance 2, execute the installer, and fill in the node configuration as follows:



5. When prompted, log in to the TSM and you will see a message showing that node 2 was added to the cluster. Click on **`Continue`**. Take a look at the **`Pending Changes`** and **`Apply Changes`** tabs and click on **`Restart`** and then **`Confirm`**, as follows:

6. If you install a total of three or more nodes, you should also deploy a co-ordination service ensemble.

Note

Read more about Coordination Services and configuration processes in the second and third nodes in the online Tableau help document. There are more materials about high availability and redundancy in the online Tableau help document. https://onlinehelp.tableau.com/current/server/en-us/distrib_ha_install_3node.htm

Deploying to Tableau Server

In this recipe, we illustrate how to configure our newly configured Tableau Server for security and usability. Tableau Server is the most secure way of sharing insights at an enterprise level. Users sign in with a login and each user must have permission to view or work with content. We will build a framework for content management.

Getting ready

Have a Tableau workbook and the Tableau Server URL ready.

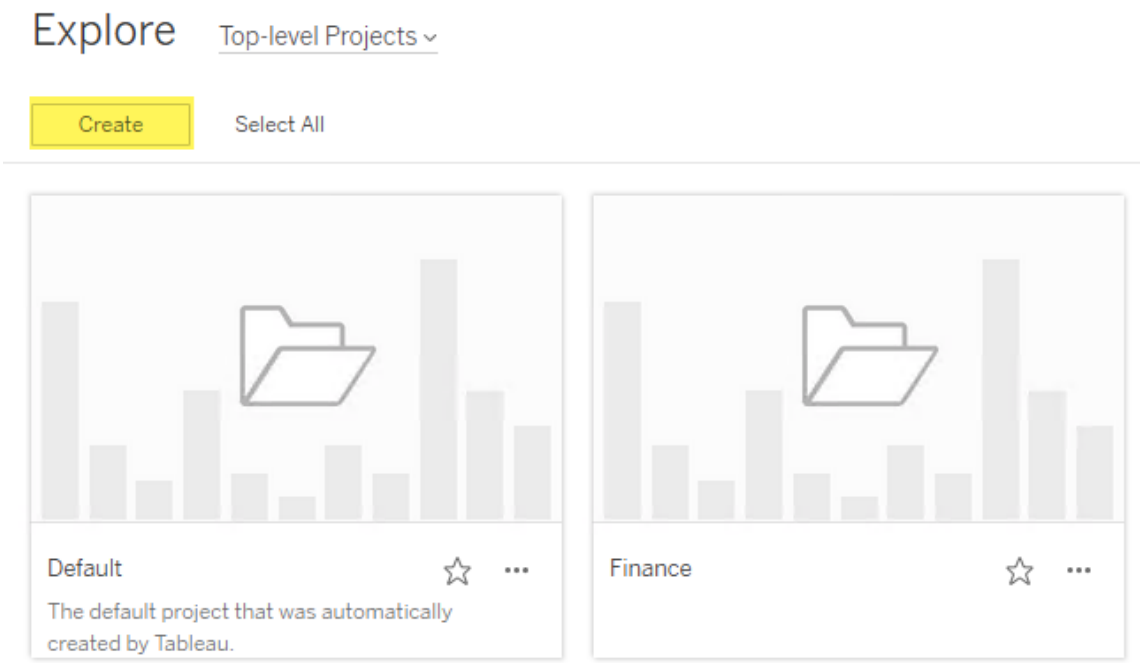
To set up the framework for your users correctly, we need to be grounded in the following three concepts:

- [**Groups**]: Sets of users who have the same access. It's best practice to set permissions at the group level and assign users to the group. Also, it is recommended that a user belong to one group as best practice for better user manageability.
- [**Projects**]: Folders and sub-folders for workbooks and data sources. It's recommended to separate content by role, function, or audience.
- [**Permissions**]: The actions that can be executed on the server and what they can impact.

How to do it..

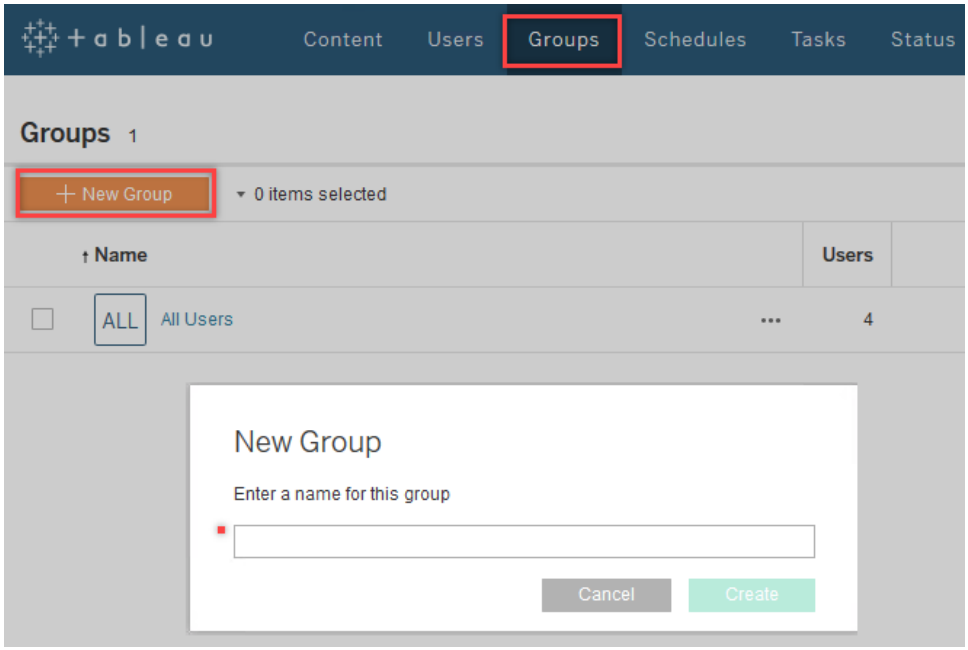
In this recipe, we will create some projects, groups, users, and set permissions.

1. Create two new projects, one for `Marketing` and `Finance`, by navigating to **`Create`** | **`Project`**, as follows:



2. Create the following three groups for **`Finance`** as shown in the following screenshot:

- `Owner`
- `Developers`
- `Viewers`



3. We create the following five local users:

- `Lisa : Finance Site Administrator Creator`
- `Mike : Finance Explorer(publish)`

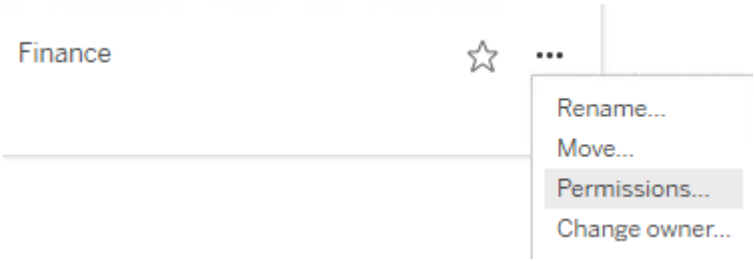
- `Eric : Finance Explorer(publish)`
- `John : Finance Viewer`
- `Satu : Finance Viewer`

Site role	View/Interact	Subscriptions & Alerts	Web Edit	Publish Views	Create Data Sources
Creator	✓	✓	✓	✓	✓
Explorer (can publish)	✓	✓	✓	✓	
Explorer	✓	✓	✓		
Viewer	✓	✓			
Unlicensed					

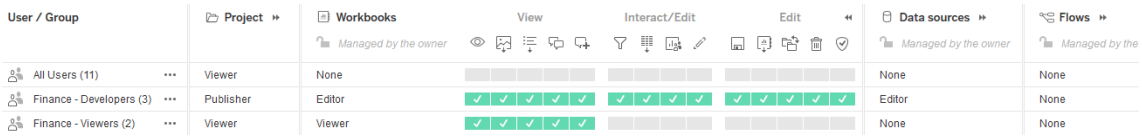
Server administrators can manage Server and manage/create Sites

Site administrators can manage a Site

4. Add those users to their respective groups. We are considering developers are those who can publish views and data source.
5. Within the **`Content`** page, add the groups to their respective projects by clicking on the ellipse and selecting **`Permissions`**, as follows:

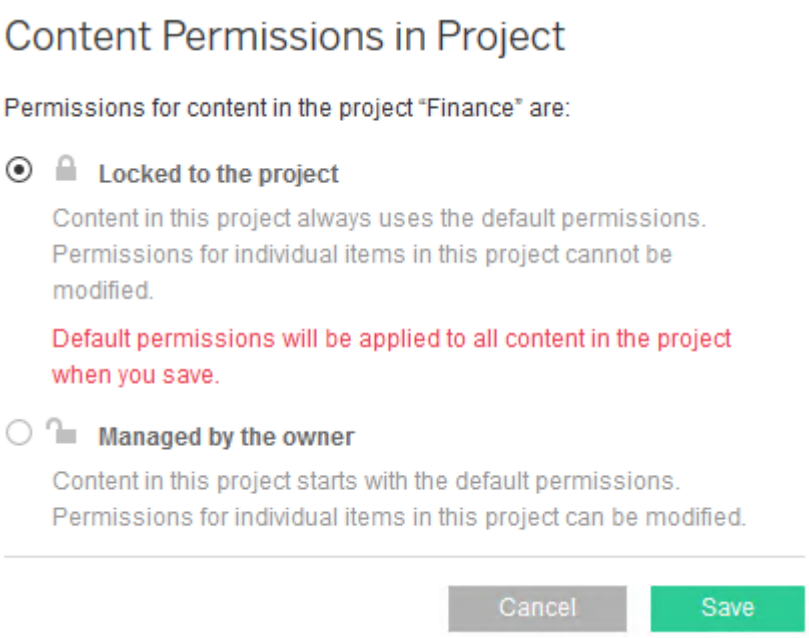


6. Add the groups to the projects and assign each group with the permissions roles. These roles act as a template to help make setup easier. To see each capability, expand the section. The permissions we set allow the content developers to manage all assets in their project, whereas the viewers can only view what's been published, as shown in the following screenshot:



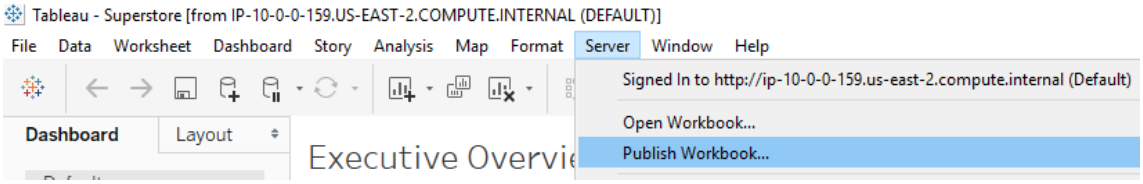
New in 2019.1, we have the ability to manage Tableau Prep flows via Tableau Prep Conductor.

7. Lock the permissions so that content publishers cannot deviate from the default permissions we set on the server, as follows:

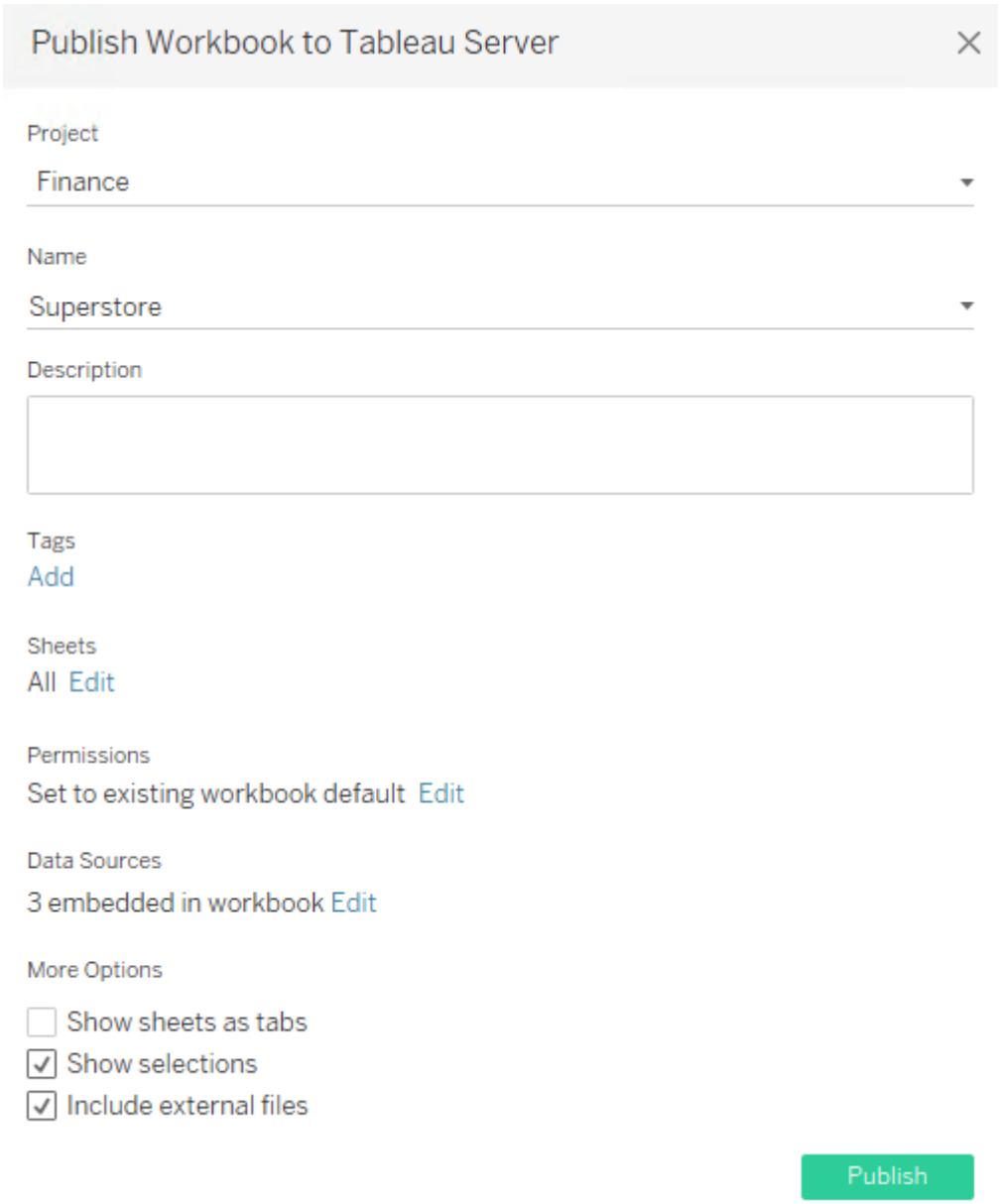


Because we made these test users, we can try publishing to the server using one account with those permissions; we can also try publishing to the server using an account without publishing access.

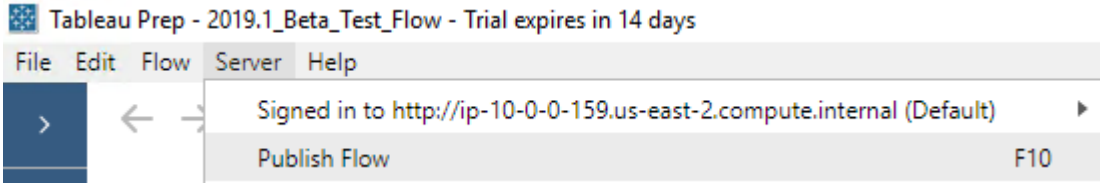
8. In Tableau Developer, open the `Superstore` sample workbook and go to the `Server` menu and select `Publish Workbook...` as follows:



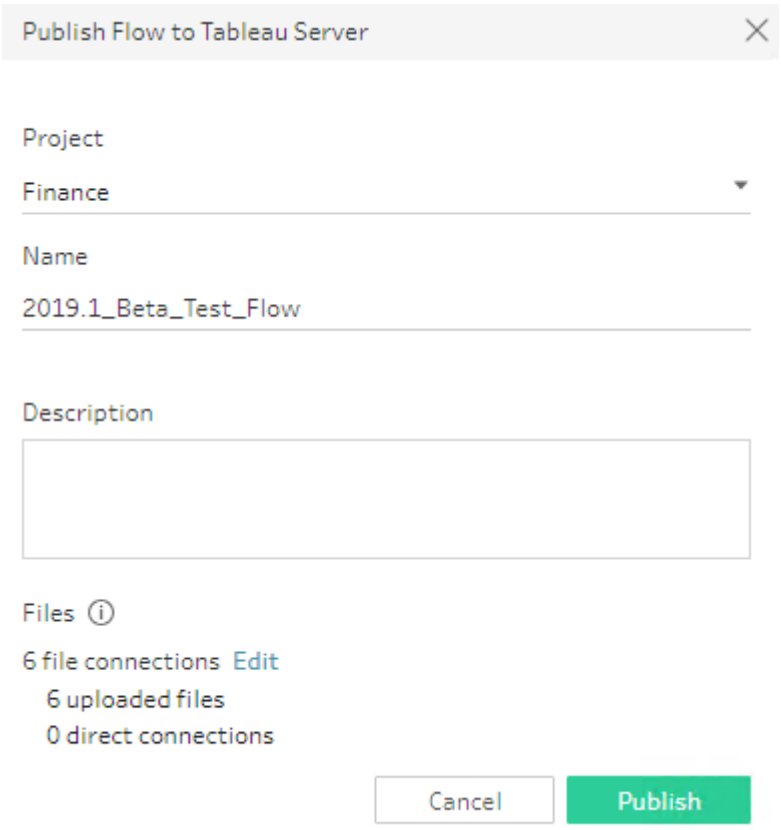
9. Publish this to the `Finance` project, as follows:



10. In Prep, open a sample flow from the beta site and go to the **`Server`** menu and select Publish Flow, as follows:



11. Publish this to the **`Finance`** project, as follows:



12. Log into the server to access the published content. Try this as different users.

How it works...

We created a project for our `Finance` team and created groups based on the activities of the users. In our prototype exercise, we created users and added these users to each group. Next, we assigned permission templates to the groups at the project level. Finally, we locked the permissions to ensure our model stays intact. We illustrated that the permissions worked by uploading one of the sample workbooks and sample flows to the server.

There's more...

There are some items to be aware of for permissions. You can only assign permissions to content. Individual permissions will take priority over any group permissions. Every user is added in the All Users group. It's important to delete this group so it doesn't cause conflicts with the other group permissions.

Deploying Tableau Server on Linux using AWS

For a long time Tableau Server was available only for Windows platforms. But with the Tableau version 10.5 there was an introduction of Tableau Server on Linux. Basically, now you have more flexibility with Tableau Server. Some organizations prefer Windows, whereas others consider only Linux. In addition, it depends on the culture of the organization and the available resources to maintain and support Linux or Windows servers. For sure, using a cloud environment will simplify this.

Tableau Server on Linux makes Tableau products more agnostic. Moreover, it increases the security of deployment and decreases the cost of ownership.

Getting ready

In this recipe, will will download the last available Tableau Server for Linux and deploy it in our AWS account on top of EC2 Linux Instance.

We can use the same VPC that we used for the Windows recipe, or we can deploy new from scratch. In this recipe we will:

- Launch EC2 Linux instance
- Download Tableau Server for Linux distribution
- Install Tableau Server and access it via SSH Before you start, you also may refer to [*Everybody's Install Guide*], which will give you a more detailed explanation behind the server installation at https://onlinehelp.tableau.com/current/guides/everybody-install-linux/en-us/everybody_admin_intro.htm.

How to do it..

Let's start our journey. This process should be straightforward and I hope you have some Linux experience. We are going to use CLI. We can choose from Ubuntu, [**Red Hat Enterprise Linux**] ([**RHEL**]), CentOS 7, Amazon Linux 2, and Oracle Linux. You should check with the documentation before installing in order to use the last available version of Linux that supports Tableau.

For our purpose we will use RHEL, but you can choose any other Linux distribution. Let's do it:

1. Log in to AWS and go to the EC2 dashboard. In the previous recipe, we created VPC and a security group that we will use for this recipe as well.
2. Now, we need to launch EC2 with RHEL 7. Go to the Amazon EC2 Console and under `Create Instance`, click on `Launch Instance`.
3. Select an [**Amazon Machine Image**] ([**AMI**]) that will meet your performance requirements. In our case, I will use `m4.xlarge`. It has 16 GB RAM and it should be enough for learning purposes. You can refer to Tableau recommendation documentation at https://onlinehelp.tableau.com/current/server-linux/en-us/ts_aws_virtual_machine_selection.htm.

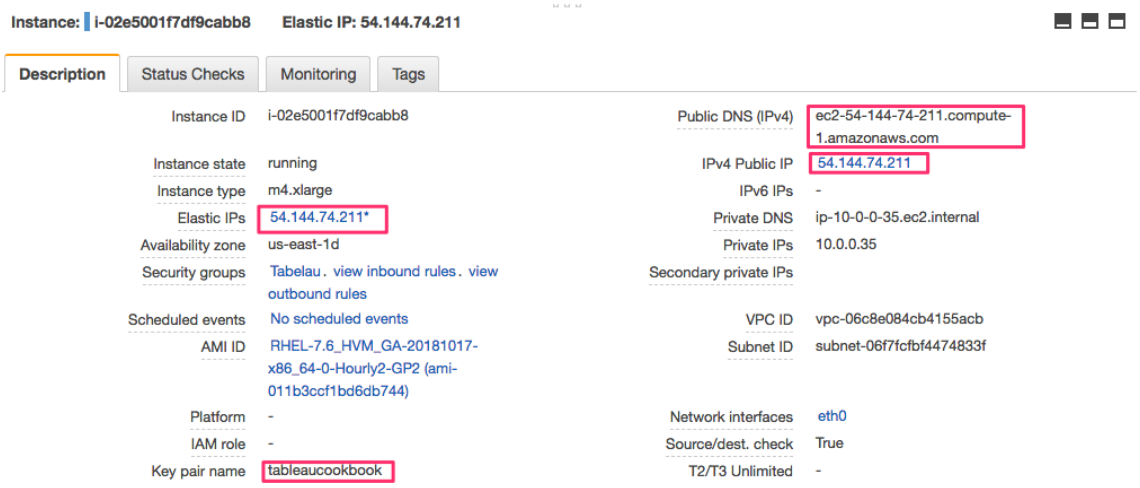
Note

We should create EC2 in the same region as we created VPC.

4. Then we should configure the instance details, as follows:

`Network`	`My_First_VPC`
`Storage Size`	`100 GiB`
`Security Group`	`my_security_group`

5. Then click on `Launch`. It will allow you to create a key pair or use an existing one. We need this for SSH to our Linux box, where we can download and install Tableau Server.
6. Let's create a static Elastic IP for our EC2 instance. We should navigate to the Amazon VPC console, go to `Elastic IP,` and click `Allocate new address`. Use all settings by default. Then, we need to associate our new Elastic IP with our EC2 instance. As a result, our EC2 instance got new IPs that we will use for connection to the host. You can refer to the following screenshot for clarity:



7. Next, we should SSH to our Linux box. If you are using macOS or Linux, then you can use CLI, but if you are using Windows, you should use Putty. In our case, we are using macOS and will open Terminal. We need to use our private key and execute the following commands:

```
cd ~/.ssh
chmod 400 tableaucookbook.pem
ssh -i ~/.ssh/tableaucookbook.pem ec2-user@ec2-54-144-74-211.compute-1.amazonaws.com
```

Here the following commands are elaborated as follows:

- `ec2-user` : This represents the default user
- `tableaucookbook.pem` : This is from our key pair
- `ec2-54-144-74-211.compute-1.amazonaws.com` : This represents the public DNS of our EC2 instance Now, we can install Tableau Server.

8. Let's download the installation file from Tableau Server. We could use the last available version of Tableau Server 2019.x. From the CLI, execute the following commands:

```
sudo yum update
sudo yum install wget
wget "https://downloads.tableau.com/esdalt/2018.3.2/tableau-server-<tableau
version>.x86_64.rpm"
```

Here, you replace `<tableau version>` in order to specify the version of Tableau Server, for example 2019-1-1.

9. Then we can install Tableau Server. Execute the following command:

```
sudo yum install tableau-server-<tableau version>.x86_64.rpm
```

This will install you server very fast, much faster and easier than on the Windows platform!

10. The next step is the same as usual. We should initialize `tsm` and `accepteula`, as follows:

```
cd /opt/tableau/tableau_server/packages/scripts.<code version>/
sudo ./initialize-tsm --accepteula
```

It will create the `tsmadmin` group for admin authorization, create a Tableau user account, and set permissions. In addition, it will add our Linux `ec2-user` to the Tableau security groups `tableau` and `tsmadmin`. In addition, it initializes a web interface for the GUI and a REST API, and will provide a link to it. But this link will have private IP and we won't be able access it from the internet. We should use a public DNS instead.

11. In addition, we should create password for our `ec2-user`, as follows:

```
sudo passwd ec2-user
```

The preceding command will prompt us to enter a new password for the user. If you don't know what the admin user is, you can run the following command:

```
grep tsmadmin /etc/group
tsmadmin:x:994:ec2-user
```

We've created a password for this user and may use it for authentication.

12. Let's activate **`Tableau Server Trial`** and log in to TSM. Before you should re-log in to the EC2, as follows:

```
tsm login -u ec2-user
tsm licenses activate -t
```

13. Next we should register Tableau Server by generating JSON and send it to Tableau. In addition, we will add text editor for Linux: `nano`, as follows:

```
sudo yum install nano
tsm register --template > /tmp/tableaucookbook.json
```

Fill the file with your information, as follows:

```
{
 "zip" : "V1R2P5",
 "country" : "Canada",
 "city" : "Victoria",
 "last_name" : "Anoshin",
 "industry" : "Dmitry",
 "eula" : "yes",
 "title" : "Boss",
 "phone" : "2508919300",
 "company" : "Amazon",
 "state" : "BC",
 "department" : "Engineering",
 "first_name" : "Dmitry",
 "email" : "dmitry.anoshin@example.com"
 }
```

14. Then execute the following command:

```
tsm register --file /tmp/tableaucookbook.json
```

You should get the message: `Registration Complete`.

15. Then we will configure `Identity Store` in order to specify the authentication method. We will use the `local` authentication. Let's create the script with config, as follows:

```
sudo nano /tmp/auth.json
{
"configEntities":{
"identityStore": {
  "_type": "identityStoreType",
  "type": "local"
  }
 }
}
```

16. Then we will import setting to Tableau, as follows:

```
tsm settings import -f /tmp/auth.json
```

If you need any other option, you can check templates here:

https://onlinehelp.tableau.com/current/server-linux/en-us/entity_identity_store.htm {.ulink}

In addition we can configure SMTP (mail server), SSL, and so on. Moreover, we can use the TSM GUI and enter all these settings much more simply.

17. Let's apply changes, as follows:

```
tsm pending-changes apply
Starting deployments asynchronous job.
Job id is '1', timeout is 10 minutes.
6% - Retrieving the topology to deploy.
13% - Retrieving the configuration to deploy.
20% - Validating the new topology.
26% - Determining if server needs to be started.
33% - Disabling all services.
40% - Waiting for the services to stop.
46% - Updating nodes to new topology.
53% - Waiting for topology to be applied.
60% - Updating nodes to new configuration.
66% - Disabling all services.
73% - Waiting for the services to stop.
80% - Reconfiguring services.
86% - Waiting for services to reconfigure.
93% - Enabling all services.
100% - Waiting for the services to start.
Successfully deployed nodes with updated configuration and topology version.
```

18. Then initialize the server by executing the following command line:

```
tsm initialize --start-server --request-timeout 1800
```

This process will take a while.

19. The final step is to add initial `Administrator` account. We will use `tabmcd` for this purpose, as follows:

```
tabcmd initialuser --server localhost:80 --username 'Administrator'
```

It will ask for passwords. We should get the following output:

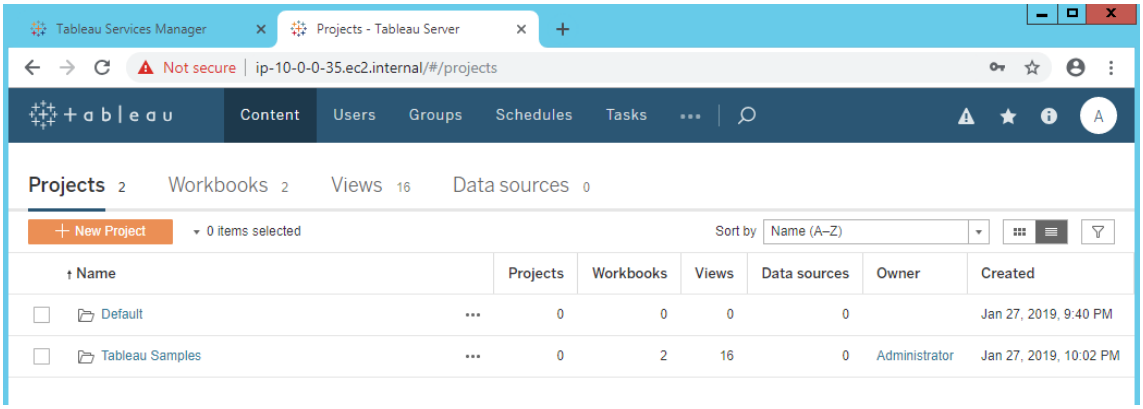
```
===== redirecting to http://localhost/auth
===== Signed out
===== Creating new session
===== Server: http://localhost:80
===== Username: Administrator
===== Connecting to the server...
===== Signing in...
===== Succeeded
```

As a result, we've got the `Administrator` account for Tableau Server and we can access it through the web.

20. Before moving to another topic, I want to highlight internal system reports for Tableau Server. Click on **`Status`** and scroll down. This list of reports is created on top of Tableau Repository based on PostgreSQL. You can learn more about reports here: https://onlinehelp.tableau.com/current/server/en-us/adminview_bucket.htm

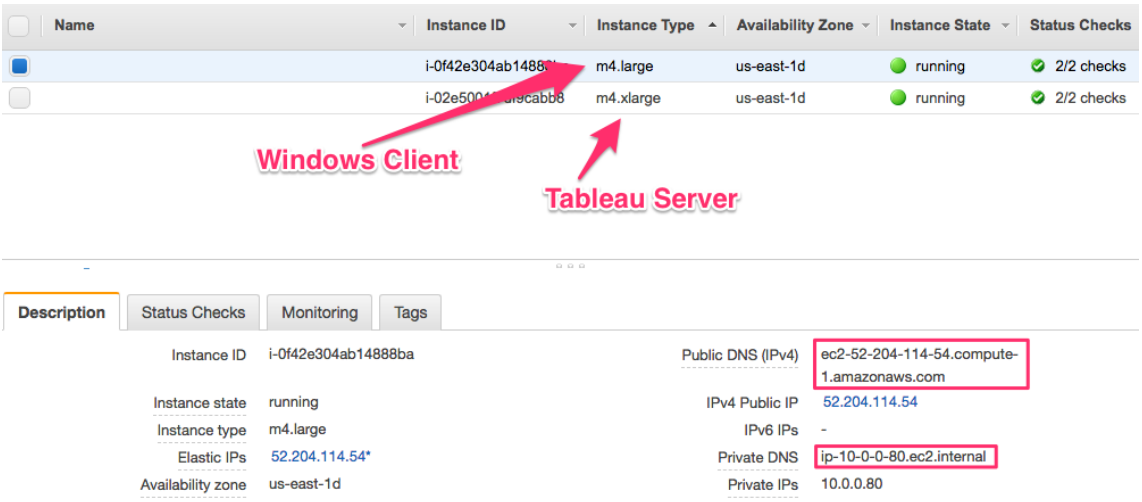
How it works...

We just finished the installation of Tableau Server on Linux. In order to access Tableau Server, we will go to the Windows EC2 client, open a browser, and type Private DNS of Linux EC2, as shown in the following screenshot:



Here, http://ip-10-0-0-35.ec2.internal/#/projects is the private DNS of our server. In addition, we have a public DNS that can be used for accessing the Tableau Server, but we should make additional network settings.

In the following screenshot, you can see my current AWS account:



I highlighted the Public DNS that I am using for the RDP client and the Private DNS for internal usage, as well as a private IP.

There's more...

For the production version, you should consider using [**Elastic Load Balancer**] ([**ELB**]) that will serve as an end point for the Tableau Server. In addition, you can upload SSL certificates to ELB and make your connection secure. Finally, you can use the ELB logging option and collect web logs.

You can run Tableau Server on Linux with a distributed environment. There is good example of it at Tableau documentation, which can be found here:

https://onlinehelp.tableau.com/current/server-linux/en-us/ts_aws_multiple_server.htm.

Getting started with Tabcmd

Tabcmd is great tool and it is the core component of your Business Intelligence solution, because it will help you with automation and integration with ETL and Data Warehouse. In this recipe, we will install Tabcmd on Windows and will learn how it can work with Tableau Server.

There are some common use case scenarios:

- Refresh Extract
- Generate Tableau Reader Book
- Generate Dashboards in PDF and upload to Shared drive or Amazon S3
- Export data from Tableau Server workbook and save in CSV or XLSX formats You can think of Tabcmd as not only an administration tool, but also a kind of a self-service tool that may be used by end users for scheduling their dashboards, and so on.

Getting ready

We will download Tabcmd for Windows machines. In this recipe, I will use a Windows machine that is existing in the same VPC and will act as a client machine. It means that Linux EC2 and Windows EC2 are on the same network. All we need to do is adjust the security group to allow access from the Windows machine.

How to do it..

Let's download Tabcmd to our Windows EC2 machine and connect Tableau Server:

1. Go to Tableau Server Releases (https://www.tableau.com/support/releases/server) and download `tabcmd` for your version of the server and your desired OS. I will download `tabcmd` for the Windows client.
2. Install Tabcmd on your machine at `C:\tabcmd\` .
3. Next, we should log in to the Tableau Server. Open CMD and type the following commands:

```
cd C:\tabcmd\"Command Line Utility"
tabcmd login -s http://ip-10-0-0-35.ec2.internal -u Administrator -p Airmax86
```

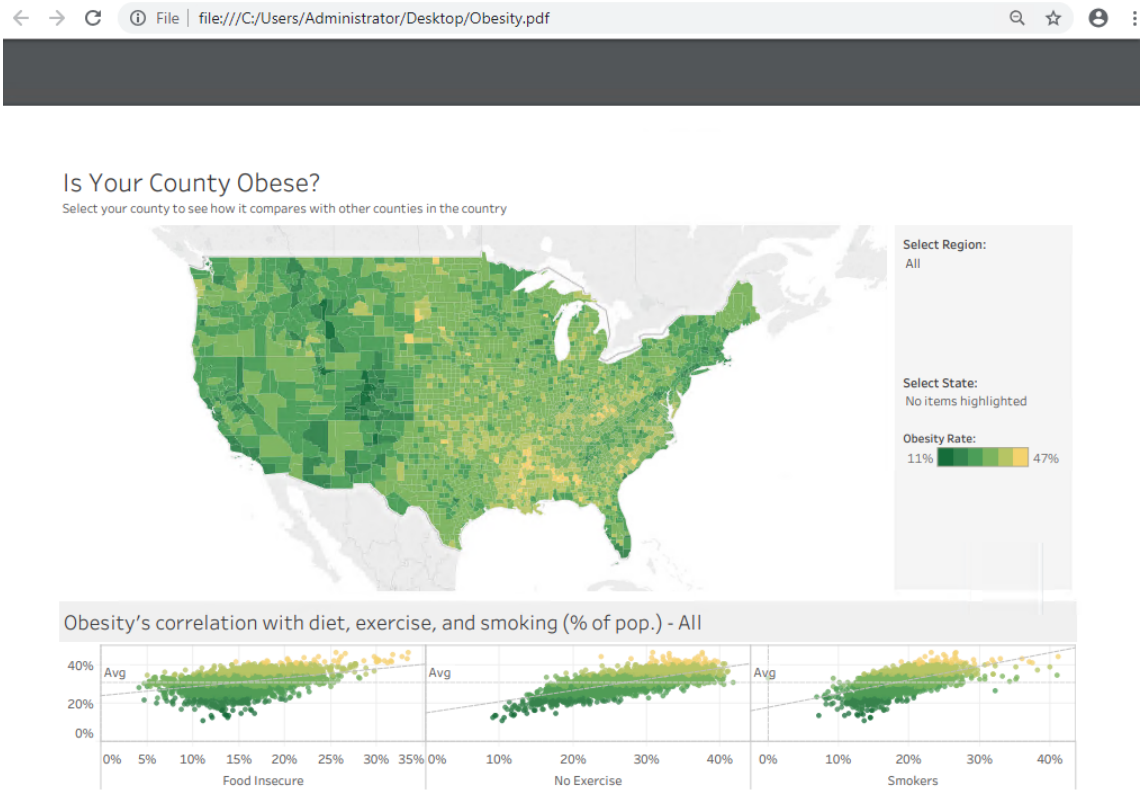
The preceding command line generates the following output:

As a result, we connected to Tableau Server via Tabcmd.

4. Next, we will execute a command to export dashboard in PDF. Our Tableau Server has sample workbooks and we will use them. Let's open Regional Workbook and Obesity Map View. We need to copy the individual views' URLs to use it in Tabcmd script. In the following image I have highlighted the part of URL that we will use for the `tabcmd` command.
5. Let's execute the following command with Tabcmd:

```
tabcmd export "Regional/Obesity" --pdf --pagelayo
 ut landscape -f "C:\Users\Administrator\Desktop\Obesity.pdf"
```

It will create a PDF file on your desktop, as follows:



In the same way, we can generate Package Workbook or Refresh Extract. We can save this command as a batch file. We need to open notepad, put in all `tabcmd` commands, including authentication and export commands, and save this for future reference or use for automation.

How it works...

Tabcmd is a special utility for Tableau Server. It is a kind of client that can trigger Tableau Server and do lots of awesome and useful things. We just installed Tabcmd on our machine and connected Server. In addition, we've got a PDF from the Server. You may do many more actions with Tabcmd. This is a link with commands:

https://onlinehelp.tableau.com/current/server/en-us/tabcmd_cmd.htm

In a real-world scenario, Tabcmd is used for automation purposes and one script will include many commands, for example, you might refresh Tableau Extract first and then export a PDF of dashboards or generate Package Workbooks.

There's more...

Building automation with Tabcmd is awesome, but it has one missing element---notifications. In the real world, I am using an open source email client for CLI: https://www.febooti.com/.

You can download this tool and try it. All you need is to insert a snippet with the Febooti command and it will deliver you an email or attachment with the file. End users love it!

Finally, you should consider using internal Windows Task Scheduler to schedule Tabcmd scripts (Batch files) that can be found here: https://en.wikipedia.org/wiki/Windows_Task_Scheduler.