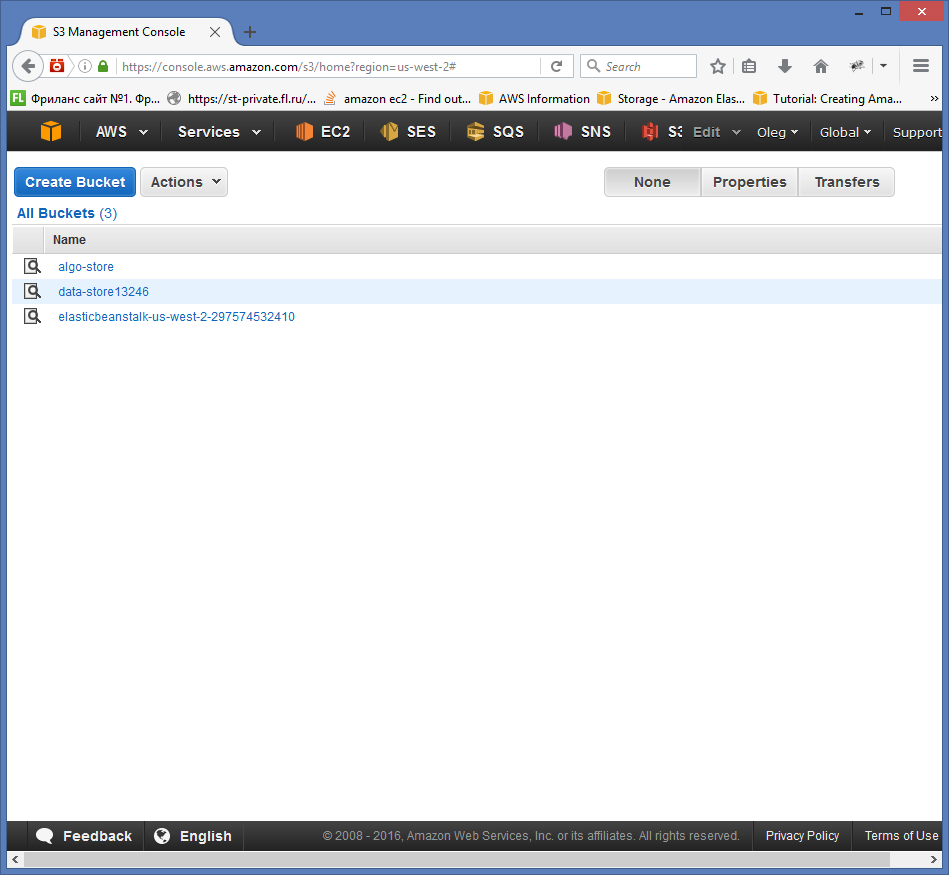
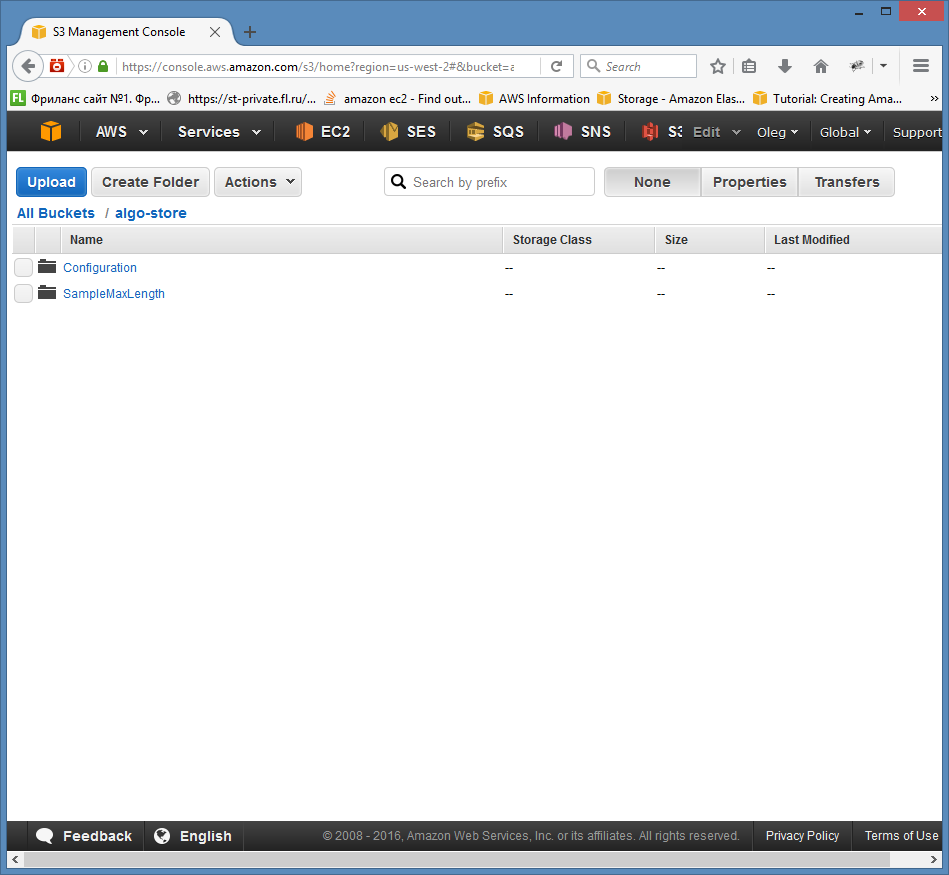
**Подготовка хранилища**

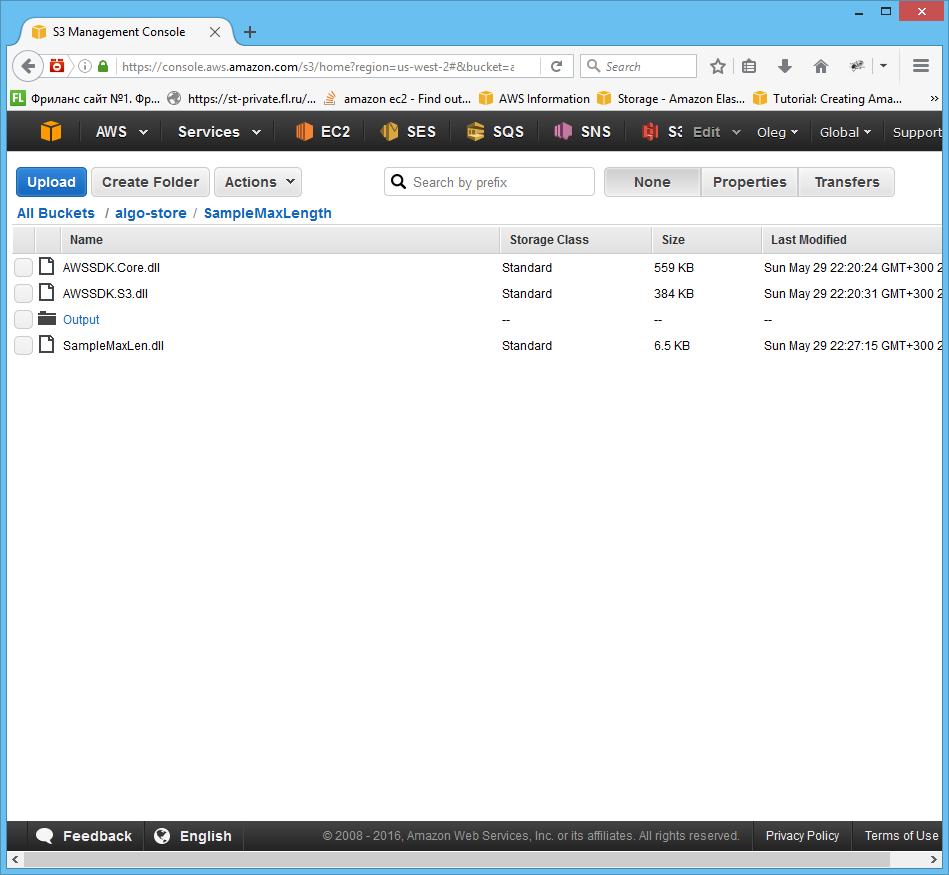
Создать 2 S3 бакета - первый для алгоритмов, второй для данных



Структура хранилища алгоритмов 

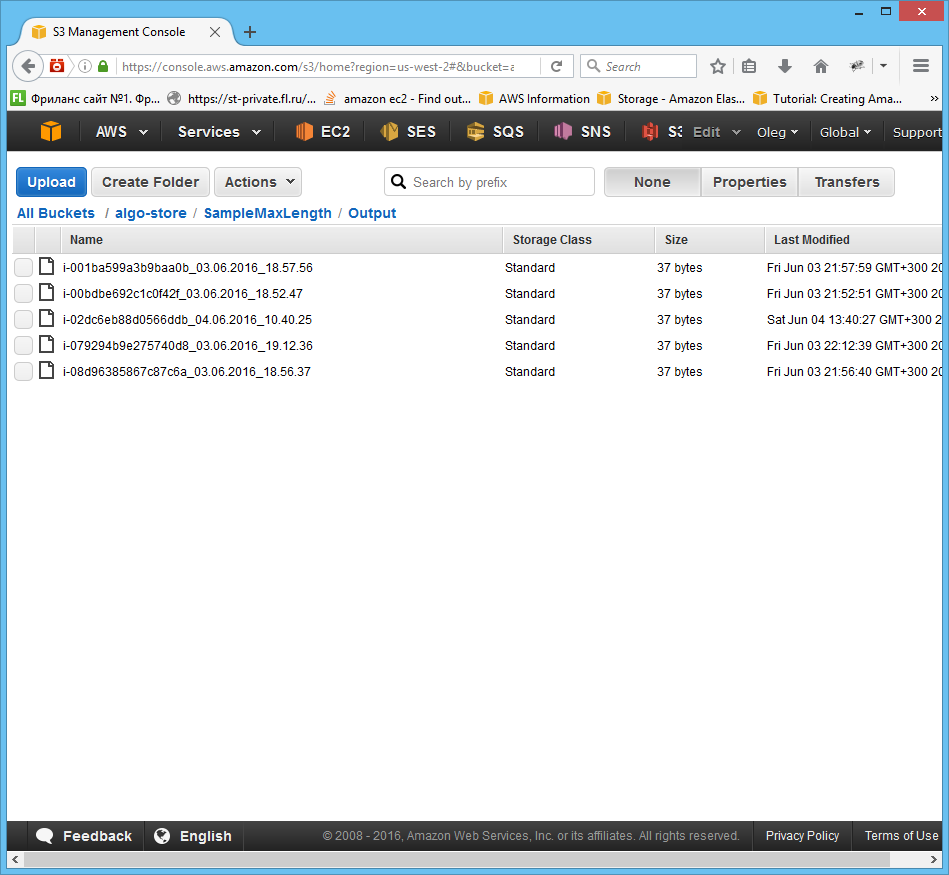
Каждый алгоритм храниться в отдельной папке. Название папки - это название алгоритма.

Так же необходимо создать пустую папку Configuration, которая будет использоваться для передачи параметров в vm.

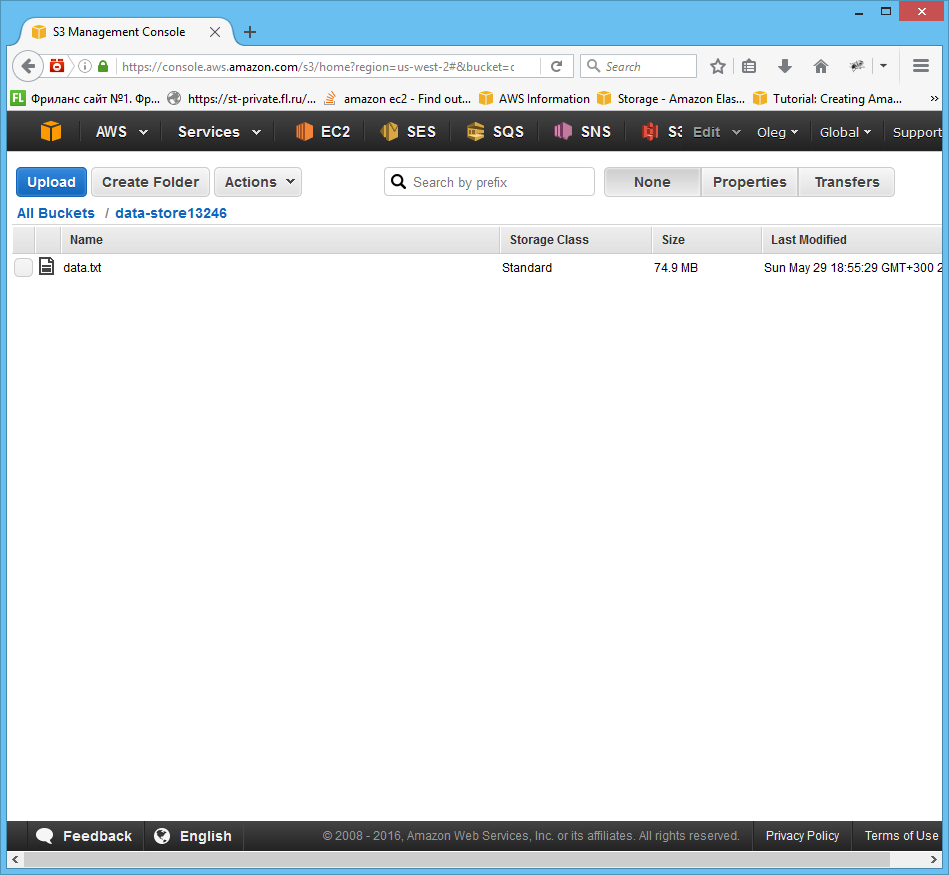


В папку с алгоритмом поместить библиотеку и зависимости.

Необходимо создать пустую папку Output для хранения результатов работы алгоритма.



В Output содержится результат работы в видел {id машины}\_дата\_время

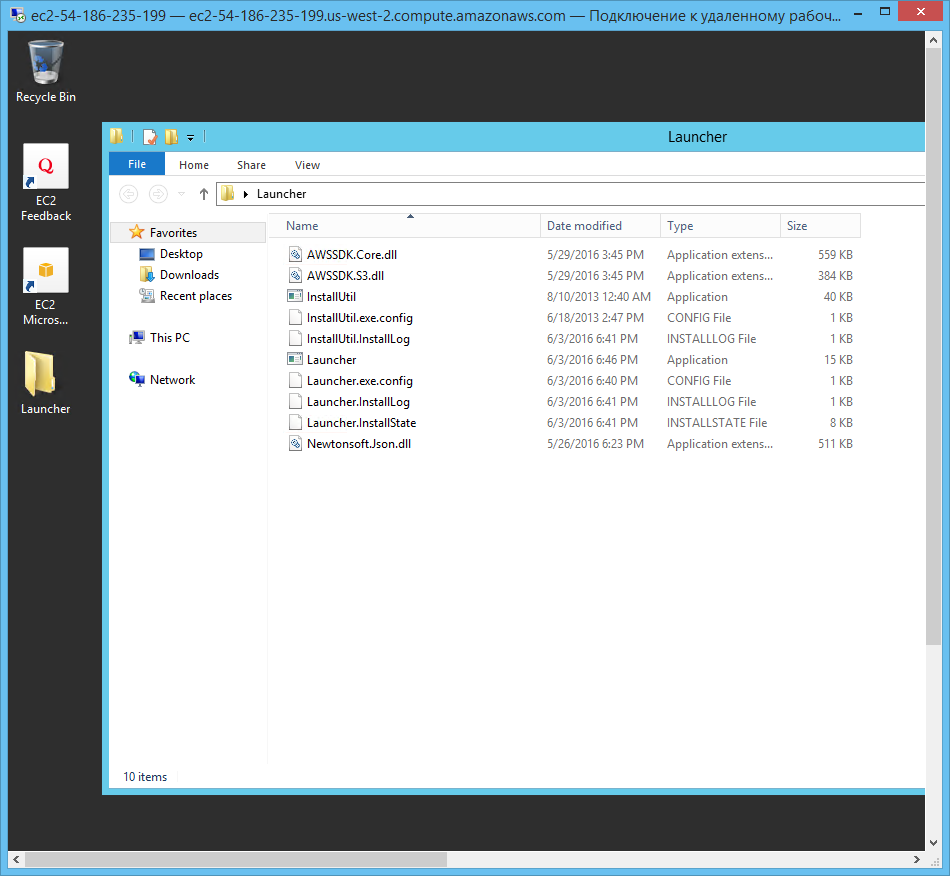


В хранилище данных поместить файл (файлы) в корневой каталог

**Подготовка службы**

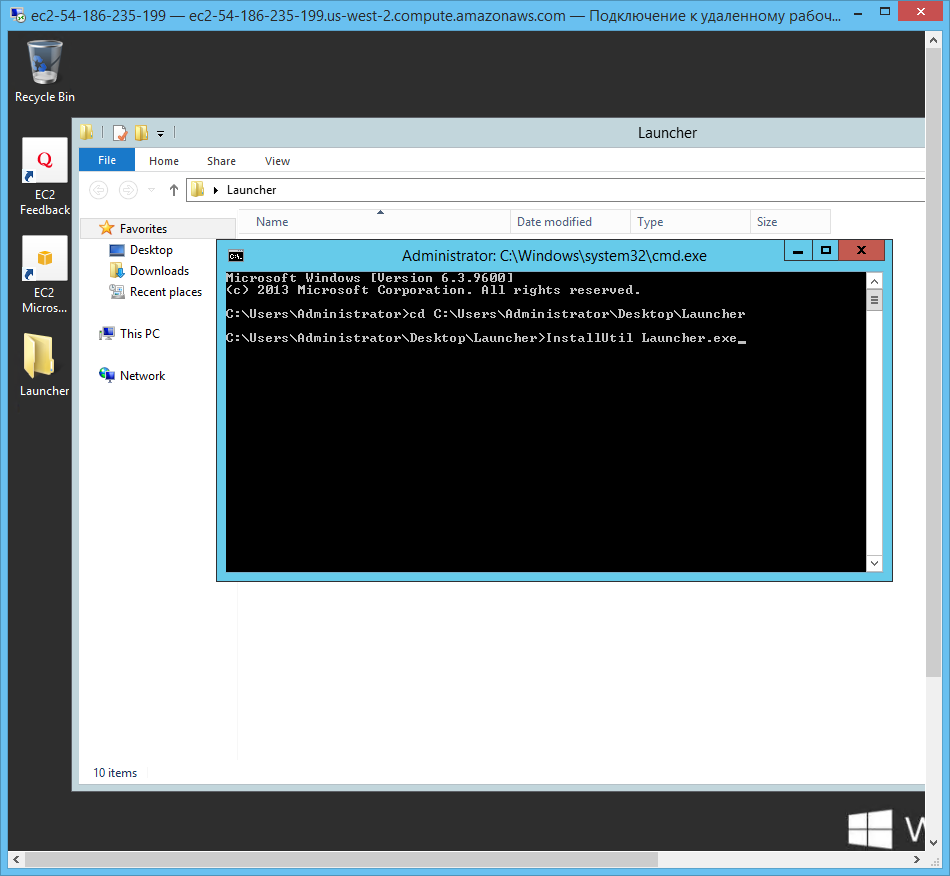
1. Создать экземпляр EC2 машины с Windows Server 2012 R2 Base

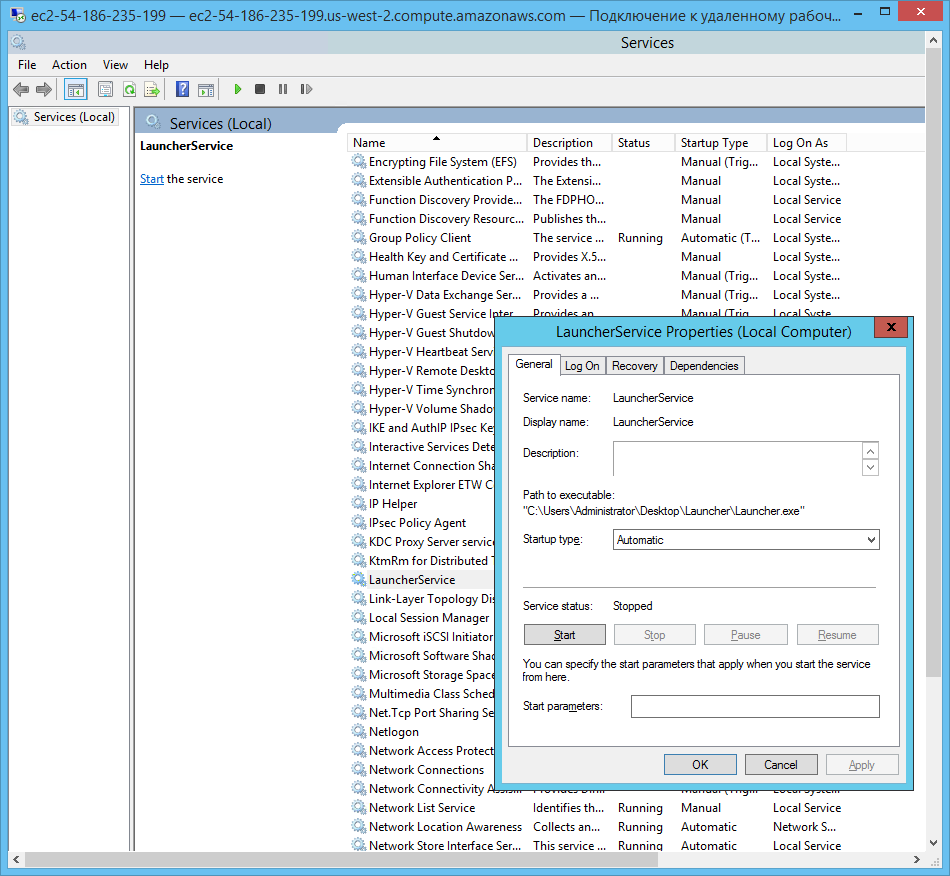
2. Подключиться через rdp.

3. Создать папку и скопировать туда службу

4. В консоли перейти в папку с программой и написать

> InstallUtil Launcher.exe



5. Служба создана, теперь необходимо ее активировать. В TaskManager -> Services -> Open Services найти LauncherService. В Properties -> Log On -> установить Local System Account, в General -> Startup type -> установить Automatic

6. В Launcher.exe.config заполнить значения в appSettings



BucketName - название хранилища алгоритмов

RegionEndpoint - название региона

LauncherRootDirectory - корневой каталог службы

CheckInputParamsInterval - интервал проверки поступления входных параметров в хранилище. Установлен в 5 секунд.

7. Виртуальная машина готова. Теперь нужно сделать с нее AMI для клонирования на другие машины. В консоли EC2 -> Instances -> Правой кнопкой по текущему экземпляру vm -> Image -> Create Image. В Image name прописать default. Больше ничего не заполнять.

**Алгоритм**

Точкой входа в алгоритм считается класс Startup, который должен содержать метод OnActivated. OnActivated может содержать (или не содержать) один параметр и возвращать object (или производный от него).

Например

public MaxLenOutput OnActivated(MaxLenInput input)  
{  
}

Где MaxLenInput может содержать строчные или числовые свойства

public class MaxLenInput  
{  
 public string DataStorage { get; set; }  
 public string FileName { get; set; }  
}

То же самое с MaxLenOutput. Свойством может быть любой сериализуемый тип

public class MaxLenOutput  
{  
 public byte Byte { get; set; }  
 public int Length { get; set; }  
 public int Index { get; set; }  
}

После завершения работы MaxLenOutput сериализируется в json и отправляется в хранилище алгоритмов в папку Output.

**Менеджер**

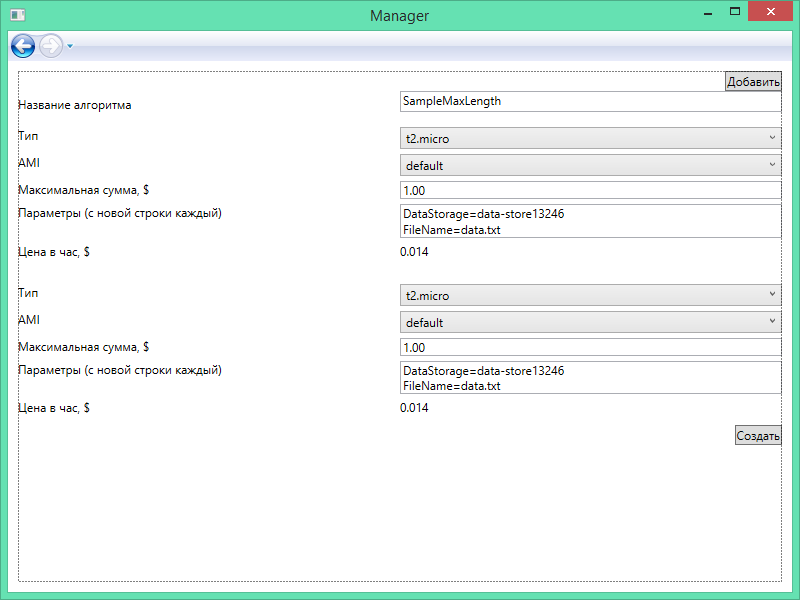
Заполнить параметры а App.config

RegionEndpoint - название региона

AlgoStorage - название хранилища алгоритмов

InstanceCheckInterval - интервал проверки экземпляров

После подготовки хранилища данных и алгоритмов, запустить менеджер, перейти на страницу создания машин



Формат параметров: ключ=значение, с новой строки каждый, где названия ключа соответствует названию свойства в входном классе данных алгоритма.

Дождаться завершения работы

